



TUGAS AKHIR - KI141502

Rancang Bangun Sistem Pemantau Cuaca (Angin) Menggunakan Mikrokontroler Arduino

**REZA DWI PUTRA
NRP 5110100033**

**Dosen Pembimbing I
Royana Muslim Ijtihadie, S.Kom., M.Kom., PhD**

**Dosen Pembimbing II
Eng. Radityo Anggoro, S.Kom., M.Sc.**

**JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2015**



UNDERGRADUATE THESES - KI141502

Weather Monitoring (Wind) System Design Using Arduino Microcontroller

**REZA DWI PUTRA
NRP 5110100033**

**Supervisor I
Royyana Muslim Ijtihadie, S.Kom., M.Kom., PhD.**

**Supervisor II
Dr.Eng. Radityo Anggoro, S.Kom., M.Sc.**

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2015**

LEMBAR PENGESAHAN
RANCANG BANGUN SISTEM PEMANTAU CUACA
(ANGIN) MENGGUNAKAN MIKROKONTROLER
ARDUINO

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Komputasi Berbasis Jaringan
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh
REZA DWI PUTRA
NRP: 5110 100 033

Disetujui oleh Dosen Pembimbing Tugas Akhir

1. Royyana Ijtihadie, S.Kom., M.Kom., Phd.
NIP: 197708242006041001 (Pembimbing 1)
2. Dr.Eng. Radityo Anggoro, S.Kom., M.Sc.
NIP: 198410162008121002 (Pembimbing 2)

SURABAYA
NOPEMBER, 2014

RANCANG BANGUN SISTEM PEMANTAU CUACA (ANGIN) MENGGUNAKAN MIKROKONTROLER ARDUINO

Nama Mahasiswa : REZA DWI PUTRA
NRP : 5110100033
Jurusan : Teknik Informatika FTIF-ITS
Dosen Pembimbing 1 : Royyana Muslim Itjihadie, S.kom.,
M.Kom., Ph.D.
Dosen Pembimbing 2 : Dr.Eng. Radityo Anggoro, S.Kom.,
M.Sc.

ABSTRAK

Kecepatan angin yang tinggi bisa menjadi ancaman bencana. Indonesia pun merupakan negara maritim yang sebagian besar wilayah negaranya adalah permukaan laut, maka Indonesia sangat rentan terhadap bencana yang diakibatkan oleh cuaca yang buruk. Salah satunya, bencana puting beliung yang diakibatkan oleh kecepatan angin yang melebihi batas. Untuk itu, diperlukan suatu alat atau aplikasi untuk memantau keadaan cuaca dengan mengukur kecepatan angin dan suhu udara.

Aplikasi pemantauan cuaca ini dirancang untuk melakukan pemantauan kecepatan angin dan suhu udara di titik yang sudah ditentukan. Data yang dibutuhkan pada aplikasi ini adalah besar kecepatan angin dan besarnya suhu udara. Dibutuhkan sensor kecepatan angin dan sensor suhu udara yang dipasang pada mikrokontroler Arduino dalam perancangan arsitektur sistem ini.

Kata kunci : Arduino, angin, pemantau, zigbee

WEATHER MONITORING (WIND) SYSTEM DESIGN USING ARDUINO MICROCONTROLLER

Student's Name : REZA DWI PUTRA
Student's ID : 5110100033
Department : Teknik Informatika FTIF-ITS
**First Advisor : Royyana Muslim Itjihadie, S.kom.,
M.Kom., Ph.D.**
**Second Advisor : Dr.Eng. Radityo Anggoro, S.Kom.,
M.Sc.**

ABSTRACT

High wind speeds can be hazardous. As a maritime country, Indonesia, which most of its area is covered in water, which makes Indonesia vulnerable to natural disasters which caused by bad weather. One of the most prevalent disaster is typhoon, which primarily caused by higher-than-average wind speed. Therefore, we need a tool or application to monitor the state of weather by measuring its wind speed and temperature.

The weather monitoring application designed to monitor wind speed and temperature at a specified point. The data needed in this application is the wind speed and the amount of temperature. It takes the wind speed sensor and temperature sensor mounted on the Arduino microcontroller in this system architecture.

Keywords: Arduino, Wind, Monitoring, Wireless Sensor Network, ZigBee

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillahirabbil'alamin, segala puji bagi Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul ***“RANCANG BANGUN SISTEM PEMANTAU CUACA (ANGIN) MENGGUNAKAN MIKROKONTROLER ARDUINO”***.

Pengerjaan Tugas Akhir ini merupakan suatu kesempatan yang sangat baik bagi penulis. Dengan pengerjaan Tugas Akhir ini, penulis bisa belajar lebih banyak untuk memperdalam dan meningkatkan apa yang telah didapatkan penulis selama menempuh perkuliahan di Teknik Informatika ITS. Dengan Tugas Akhir ini penulis juga dapat menghasilkan suatu implementasi dari apa yang telah penulis pelajari.

Selesaiannya Tugas Akhir ini tidak lepas dari bantuan dan dukungan beberapa pihak. Sehingga pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Allah SWT atas limpahan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan baik.
2. Keluarga penulis, Ibu Dindin Setyawati Yusuf dan Bapak Rudi Syafruddin Kala dan, kakak penulis Jodi Perdana Kala yang telah memberikan dukungan moral, spiritual dan material kepada penulis. Serta selalu memberikan doa yang tiada habisnya yang dipanjatkan untuk penulis.
3. Bapak Royyana Muslim, S.Kom., M.Kom, Ph.D., selaku dosen pembimbing pertama yang telah memberikan kepercayaan, dukungan, bimbingan, nasehat, perhatian yang telah diberikan kepada penulis.
4. Bapak Dr.Eng. Radityo Anggoro, S.Kom., M.Kom., M.Sc. selaku dosen pembimbing kedua, atas bimbingan, arahan, bantuan serta ide untuk menyelesaikan Tugas Akhir ini.
5. Ibu Nanik Suciati, S.Kom., M.Kom., Dr.Eng. selaku ketua jurusan Teknik Informatika ITS dan dosen wali penulis dan

segenap Bapak/Ibu dosen Teknik Informatika yang telah memberikan ilmunya kepada penulis.

6. Pak Yudi, Pak Sugeng, Mas Jumali, Mas Muryono, Mas Hari, Mas Doni dan segenap staf Tata Usaha yang telah memberikan segala bantuan dan kemudahan kepada penulis selama menjalani kuliah di Teknik Informatika ITS.
7. Mas Aditya Elektro ITS 07, Grezio, Holong, Cahya TC 09, dan Gigih TC 09 yang telah membantu penulis selama pengerjaan Tugas Akhir.
8. Sahabat karib penulis, Nabila, Archyuda, Rizkiadi, Abdu, Tegar, Panji, Fuad, Meutia, Bena, dan Harfi.
9. Kontrakan Kimochi, Rendy, Amal, Angga, Zaki, Zakki, Ditri, Adam, Iqbal, Gading, dan Azhar yang telah menjadi keluarga penulis selama di Surabaya.
10. Kontrakan T-3, Anto, Fazri, bang Aji, Candra, Agil, Teguh, bang Wely, Dimas, bang Afif, bang Mamon, bang Andre, dan bang Yogi yang telah mengenalkan penulis kepada Surabaya.
11. Gatotkaca SMA Negeri 1 Bogor atas motivasinya yang luar biasa.
12. Teman-teman TC angkatan 2010 yang selalu menjaga kebersamaan.
13. Gilang, Radite, Alief, Yanto, selaku teman seperjuangan satu bimbingan.
14. Fahry, Nabil, Naufal, Arthur, Samodro, Ibrahim, Adriyanra, Valentino, Ari, Adrie yang telah memudahkan adaptasi penulis selama di Surabaya.
15. Teman seperjuangan SW 111 Sujarwe, Afif, Bobby, Dmitri, Pras, Guruh, Varistha, Bambang, Irham, Febri, Yudha, Ilmal dan semua teman yang tidak bisa disebutkan satu per satu.
16. Teman-teman Bogor dan bidang minat NCC terima kasih atas kebersamaannya selama ini.
17. Keluarga bapak Asep Cahyana Sanusi atas motivasinya secara tidak langsung.
18. Dan semua pihak yang telah membantu Tugas Akhir penulis.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan. Sehingga dengan kerendahan hati, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depan.

Surabaya, Nopember 2014

DAFTAR ISI

LEMBAR PENGESAHAN	vii
<i>Abstrak</i>	ix
<i>Abstract</i>	xi
KATA PENGANTAR	xiii
DAFTAR ISI	xvi
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	3
1.5 Manfaat	3
1.6 Metodologi	3
1.7 Sistematika Penulisan Laporan Tugas Akhir	5
BAB II TINJAUAN PUSTAKA	7
2.1 <i>Microcontroller</i> Arduino	7
2.2 Sensor Anemometer	9
2.3 <i>Weather Meters</i>	9
2.4 Sensor Suhu Udara (DHT22)	9
2.5 Protokol ZigBee	10
2.6 XBee <i>Module</i>	11
2.7 PHP	13
2.8 XBee <i>Shield</i>	14
2.9 Angin Puting Beliung	14
BAB III PERANCANGAN PERANGKAT LUNAK DAN KERAS	16
3.1 Deskripsi Sistem Secara Umum	16
3.2 Arsitektur Umum Sistem	16
3.3 Perancangan Basis Data	18
3.4 Perancangan Perangkat Keras	19
3.4.1 Perancangan Perangkat Keras ZigBee Koordinator	19
3.4.2 Perancangan Perangkat Keras ZigBee <i>End Device</i>	20

3.5	Diagram Alir Aplikasi	21
3.6	Diagram Alir Aplikasi Sistem.....	21
3.6.1	Diagram Alir Mendeteksi Nilai Suhu Udara	21
3.6.2	Diagram Alir Mendeteksi Nilai Kecepatan Angin	22
3.6.3	Diagram Alir <i>Microcontroller</i>	23
3.6.4	Diagram Alir Penerimaan Data Pada Komputer	24
3.6.5	Rancangan Antar Muka Aplikasi	24
	BAB IV IMPLEMENTASI	27
4.1	Lingkungan Implementasi	27
4.1.1	Lingkungan Implementasi Perangkat Keras.....	27
4.1.2	Lingkungan Implementasi Perangkat Lunak.....	27
4.2	Implementasi Perangkat Keras ZigBee Koordinator	28
4.3	Implementasi <i>Firmware</i> ZigBee Koordinator	29
4.4	Implementasi Perangkat Lunak ZigBee Koordinator	30
4.5	Implementasi Perangkat Keras ZigBee <i>End Device</i>	31
4.6	Implementasi <i>Firmware</i> ZigBee <i>End device</i>	32
4.7	Implementasi Perangkat Lunak ZigBee <i>End Device</i>	34
4.8	Implementasi Pada <i>Server</i>	34
4.9	Implementasi Memasukkan Data ke Basis Data.....	34
4.10	Implementasi Menampilkan Data pada <i>Web</i>	35
4.11	Implementasi Tampilan Grafik pada <i>Web</i>	35
	BAB V PENGUJIAN DAN EVALUASI.....	39
5.1	Lingkungan Uji Coba	39
5.1.1	Uji Coba Mendapatkan Data Kecepatan Angin dan Suhu Udara	42
5.1.2	Uji Coba Menampilkan Peringatan dan Alarm Potensi Puting Beliung.....	43
5.2	Uji Coba Performa Sistem	45
5.2.1	Uji Coba Sistem pada Lantai 4 Teknik Informatika ITS.	45
5.2.1.1	Uji Coba pada Jarak 5 Meter.....	45
5.2.1.2	Uji Coba pada Jarak 10 Meter.....	47
5.2.1.3	Uji Coba pada Jarak 15 Meter.....	49
5.2.2	Uji Coba Sistem pada Lapangan Sepak Bola ITS	50
5.2.2.1	Uji Coba Sistem pada Jarak 5 Meter	51
5.2.2.2	Uji Coba Sistem pada Jarak 15 Meter	52

5.2.2.3	Uji Coba Sistem pada Jarak 30 Meter	54
5.2.2.4	Uji Coba Sistem pada Jarak 70 Meter	55
5.2.2.5	Uji Coba Sistem pada Jarak 180 Meter	57
5.2.3	Perbandingan Berbagai Jarak dan Lokasi	58
5.2.3.1	Perbandingan Uji Coba di Lantai 4 Teknik Informatika ITS	58
5.2.3.2	Perbandingan Uji Coba di Lapangan Sepak Bola ITS	59
BAB VI KESIMPULAN DAN SARAN		62
6.1	Kesimpulan.....	62
6.2	Saran.....	62
DAFTAR PUSTAKA.....		64
LAMPIRAN		65
BIODATA PENULIS.....		70

DAFTAR GAMBAR

Gambar 2.1 Perangkat <i>microcontroller</i> arduino	8
Gambar 2.2 Sensor DHT22.....	10
Gambar 2.3 Gambar A adalah Xbee seri 1 & Gambar B adalah Xbee seri 2	13
Gambar 2.4 XBee <i>shield</i>	14
Gambar 3.1 Arsitektur Sistem Secara Umum	17
Gambar 3.2 Perancangan Basis Data	18
Gambar 3.3 Perancangan Perangkat ZigBee Koordinator	19
Gambar 3.4 Perancangan Perangkat ZigBee <i>End Device</i>	20
Gambar 3.5 Perancangan Diagram Alir Sistem	21
Gambar 3.6 Diagram Alir Mendeteksi Nilai Suhu Udara	22
Gambar 3.7 Diagram Alir Mendeteksi Nilai Kecepatan Angin	23
Gambar 3.8 Diagram Alir <i>Microcontroller</i>	23
Gambar 3.9 Diagram Alir Penerimaan Data Pada Komputer	24
Gambar 3.10 Rancang Bangun <i>Web</i> Aplikasi.....	25
Gambar 4.1 Perangkat Keras yang Dibutuhkan ZigBee Koordinator	28
Gambar 4.2 ZigBee Koordinator Setelah di Rakit	29
Gambar 4.3 Pemasangan <i>Firmware</i> ZigBee Koordinator Dengan Menggunakan X-CTU	30
Gambar 4.4 Implementasi Perangkat Lunak ZigBee Koordinator	31
Gambar 4.5 Perangkat ZigBee <i>End Device</i>	32
Gambar 4.6 ZigBee <i>End Device</i> Setelah Dirakit	32
Gambar 4.7 Pemasangan <i>Firmware</i> pada ZigBee <i>End Device</i> ...	33
Gambar 4.8 Implementasi Perangkat Lunak ZigBee <i>End Device</i>	34
Gambar 4.9 Implementasi Memasukkan Data ke Basis Data	35
Gambar 4.10 Implementasi Menampilkan <i>Chart</i> pada <i>Web</i>	35
Gambar 4.11 Antarmuka node 1	36
Gambar 4.12 Antarmuka rekap nilai harian.....	37

Gambar 5.1 Suasana uji coba ZigBee <i>end device</i> di lantai 4 Teknik Informatika ITS	40
Gambar 5.2 Suasana uji coba pada ZigBee koordinator di lantai 4 Teknik Informatika ITS	40
Gambar 5.3 Suasana uji coba pada ZigBee <i>end device</i> di lapangan sepak bola ITS	41
Gambar 5.4 Suasana uji coba pada ZigBee koordinator di lapangan sepak bola ITS	41
Gambar 5.5 Uji coba mendapatkan data kecepatan angin dan suhu udara	43
Gambar 5.6 Grafik antarmuka dengan kotak peringatan dan alarm	44
Gambar 5.7 Visualisasi jarak ZigBee <i>end device</i> dan koordinator	46
Gambar 5.8 Visualisasi jarak ZigBee <i>end device</i> dan koordinator	47
Gambar 5.9 Visualisasi jarak ZigBee <i>end device</i> dan koordinator	49
Gambar 5.10 Visualisasi jarak ZigBee <i>end device</i> dan koordinator	51
Gambar 5.11 Visualisasi jarak ZigBee <i>end device</i> dan koordinator	52
Gambar 5.12 Visualisasi jarak ZigBee <i>end device</i> dan koordinator	54
Gambar 5.13 Visualisasi jarak ZigBee <i>end device</i> dan koordinator	55
Gambar 5.14 Visualisasi jarak ZigBee <i>end device</i> dan koordinator	57
Gambar 5.15 Grafik hasil pengujian <i>packet loss</i> lantai 4 Teknik Informatika ITS	59
Gambar 5.16 Grafik hasil pengujian <i>packet loss</i> lapangan sepak bola ITS	60

DAFTAR TABEL

Tabel 2.1 <i>Datasheet</i> sensor DHT22	10
Tabel 2.2 Perbandingan antara ZigBee, Bluetooth dan WiFi	11
Tabel 2.3 Perbandingan Xbee seri 1 dan Xbee seri 2	12
Tabel 5.1 Skenario uji coba mendapatkan data kecepatan angin dan suhu udara.....	42
Tabel 5.2 Skenario uji coba menampilkan peringatan dan alarm potensi puting beliung	44
Tabel 5.3 Skenario uji coba pada lantai 4 Teknik Informatika...	45
Tabel 5.4 Data uji coba jarak 5 meter.....	46
Tabel 5.5 Data uji coba jarak 10 meter.....	48
Tabel 5.6 Data uji coba jarak 15 meter.....	49
Tabel 5.7 Skenario uji coba pada lapangan sepak bola ITS	50
Tabel 5.8 Data uji coba jarak 5 meter.....	51
Tabel 5.9 Data uji coba jarak 15 meter.....	53
Tabel 5.10 Data uji coba jarak 30 meter.....	54
Tabel 5.11 Data uji coba jarak 70 meter.....	56
Tabel 5.12 Data uji coba jarak 180 meter.....	57
Tabel 5.13 Hasil uji <i>packet loss</i> lantai 4 Teknik Informatika ITS	58
Tabel 5.14 Hasil pengujian <i>packet loss</i> di lapangan sepak bola ITS	60

BIODATA PENULIS



Reza Dwi Putra, biasa dipanggil Reza lahir di kota Bogor, Jawa Barat pada 15 Juli 1991. Penulis adalah anak kedua dari dua bersaudara. Penulis menempuh pendidikan SDI Bina Insani (1997-2003), SMPN 4 Bogor (2003-2006), SMAN 1 Bogor (2006-2009). Penulis kemudian melanjutkan pendidikan di Teknik Informatika ITS pada tahun 2010 melalui jalur PKM-Kemitraan. Penulis dapat dihubungi melalui alamat *e-mail* di

rezadwiputra@rocketmail.com

BAB I

PENDAHULUAN

Pada bab pendahuluan akan dijelaskan mengenai beberapa hal dasar dalam Tugas Akhir yang meliputi latar belakang, rumusan masalah, tujuan dan manfaat pembuatan Tugas Akhir, serta metodologi dan sistematika penulisan buku Tugas Akhir.

1.1 Latar Belakang

Indonesia memiliki wilayah yang dilewati oleh garis khatulistiwa, yaitu Pontianak. Karena itu, kecepatan angin di Indonesia ini pun menjadi salah satu yang tertinggi berdasarkan hasil statistik BMKG. Kecepatan angin yang tinggi bisa menjadi ancaman bencana. Indonesia pun merupakan negara maritim yang sebagian besar wilayah negaranya adalah permukaan laut, maka Indonesia sangat rentan terhadap bencana yang diakibatkan oleh cuaca yang buruk. Salah satunya, bencana puting beliung yang diakibatkan oleh kecepatan angin yang melebihi batas. Untuk itu, diperlukan suatu alat atau aplikasi untuk memantau keadaan cuaca dengan mengukur kecepatan angin dan suhu udara.

Mikrokontroler Arduino Uno merupakan sebuah mikrokontroler ATMEGA328 yang memiliki *flash memory* dan RAM. Mikrokontroler ini dapat dihubungkan dengan sensor yang dapat mendeteksi kondisi sekitarnya, salah satunya sensor kecepatan angin. Aplikasi pemantauan kecepatan angin ini dirancang untuk melakukan pemantauan kecepatan angin untuk mendeteksi dini potensi bencana alam akibat cuaca buruk di titik yang sudah ditentukan. Data yang dibutuhkan pada aplikasi ini adalah besar kecepatan angin dan besar suhu. Dibutuhkan sensor kecepatan angin dan suhu DHT22 dalam perancangan arsitektur sistem ini. Data kecepatan angin akan di kirim oleh mikrokontroller Arduino kepada aplikasi secara berkala. Tujuannya agar data kecepatan angin dan suhu udara dapat dibaca dan diolah dalam sebuah aplikasi web. Untuk mengirim data ke aplikasi, digunakan sebuah perangkat bernama XBee. Pengiriman data didahului

dengan melakukan koneksi ke aplikasi. Proses selanjutnya adalah melakukan pengiriman data dengan XBee modul. Aplikasi nantinya digunakan oleh *end user*.

Untuk mendapatkan data kecepatan angin, mikrokontroller akan mengirim secara terus menerus. Ketika aplikasi mengirim data ke aplikasi, secara otomatis, aplikasi akan mengolah dan mengeluarkan hasil deteksi. Nantinya, data yang sudah diterima, akan dimasukkan dalam suatu *rule* yang berfungsi sebagai acuan apakah kecepatan angin dan suhu saat ini berpotensi menjadi bencana puting beliung atau tidak. Tujuan aplikasi ini yakni untuk membantu memonitor kondisi kecepatan angin di tempat yang rawan terjadi bencana angin puting beliung. Diharapkan dengan aplikasi ini dapat dilakukan pencegahan secara dini dari bencana puting beliung.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana cara mengukur kecepatan angin yang dihasilkan dari sensor Arduino?
2. Bagaimana cara mengolah dan mengirim data ke aplikasi?
3. Bagaimana cara menentukan potensi bencana angin puting beliung?
4. Bagaimana cara mengalibrasikan dua data yang didapat dari mikrokontroller?
5. Bagaimana cara mengolah dan menyajikan informasi pada aplikasi?

1.3 Batasan Masalah

Dari permasalahan yang telah diuraikan di atas, terdapat beberapa batasan masalah pada Tugas Akhir ini, di antaranya sebagai berikut:

1. *Microcontroller* menggunakan series arduino uno.
2. Sensor yang digunakan adalah sensor kecepatan angin dan sensor suhu.

3. Terdapat satu perangkat Xbee Shield pada tiap mikrokontroller Arduino untuk mengirimkan data pada aplikasi.
4. Perangkat Xbee hanya efektif mengirimkan data sejauh 70 meter.
5. Pengiriman data sangat terbatas jika berada didalam ruangan.
6. Aplikasi berbasis web interface dengan bahasa PHP dan JavaScript.

1.4 Tujuan

Tujuan dari pembuatan Tugas Akhir adalah membuat sebuah alat yang dapat menjadi sistem pemantauan cuaca menggunakan *microcontroller* arduino uno. Alat dapat mengetahui kondisi perubahan kecepatan angin dan suhu udara secara *realtime* sehingga dapat mengurangi jumlah korban jiwa.

1.5 Manfaat

Manfaat yang diharapkan dari pengerjaan Tugas Akhir ini adalah untuk memberikan solusi pemantauan kecepatan angin di area yang rawan terjadi bencana dengan menggunakan sensor berbasis mikrokontroler Arduino.

1.6 Metodologi

Tahapan-tahapan yang dilakukan dalam pengerjaan Tugas Akhir ini adalah sebagai berikut:

1. Penyusunan proposal Tugas Akhir.
Tahap awal untuk memulai pengerjaan Tugas Akhir adalah penyusunan proposal. Pada proposal tersebut dijelaskan secara garis besar tentang membaca data dari *microcontroller* dan metode pengiriman data menggunakan XBee.
2. Studi literatur

Tahap ini merupakan tahap pengumpulan informasi yang diperlukan untuk pengerjaan Tugas Akhir sekaligus mempelajarinya. Mulai dari pengumpulan literatur, diskusi, pemahaman topik tentang kecepatan angin dan suhu udara, Perancangan alat, modul operational arduino uno, modul operational XBEE Shield dan dokumentasi internet.

3. Perancangan sistem

Tahap ini merupakan perancangan sistem dengan menggunakan studi literatur. Dengan berbekal teori, metode dan informasi yang sudah terkumpul pada tahap sebelumnya diharapkan dapat membantu dalam proses perancangan sistem.

4. Implementasi perangkat lunak

Tahap ini merupakan implementasi rancangan alat yang telah dibuat. Tahap ini merealisasikan apa yang terdapat pada tahapan perancangan yang telah dibuat sebelumnya sehingga menjadi sebuah alat pemantau kecepatan angin yang sesuai dengan apa yang telah direncanakan.

5. Pengujian dan evaluasi

Aplikasi akan diuji setelah selesai diimplementasikan menggunakan skenario yang sudah dipersiapkan. Pengujian dan evaluasi akan dilakukan dengan melihat kesesuaian dengan perencanaan. Dengan melakukan pengujian dan evaluasi dimaksudkan juga untuk mengevaluasi jalannya sensor, mencari masalah yang mungkin timbul dan mengadakan perbaikan jika terdapat kesalahan.

6. Penyusunan buku Tugas Akhir.

Pada tahap ini disusun laporan Tugas Akhir sebagai dokumentasi pelaksanaan Tugas Akhir, yang mencakup seluruh konsep, teori, implementasi, serta hasil yang telah dikerjakan.

1.7 Sistematika Penulisan Laporan Tugas Akhir

Buku Tugas Akhir ini disusun dengan sistematika penulisan sebagai berikut:

BAB I. PENDAHULUAN

Bab yang berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan Tugas Akhir. Selain itu permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

BAB II. TINJAUAN PUSTAKA

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan Tugas Akhir ini.

BAB III. PERANCANGAN PERANGKAT LUNAK DAN KERAS

Bab ini berisi tentang perancangan sistem dan *flowchart* yang akan dibuat. Perancangan yang dibahas meliputi perancangan sistem, pengambilan gambar dan menyimpan wajah pada gambar, perbandingan gambar wajah antara gambar wajah yang telah tersimpan dan gambar wajah yang akan membuka kunci, dan mekanisme rangkaian perangkat keras untuk membuka kunci *solenoid*.

BAB IV. IMPLEMENTASI

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa *code* yang digunakan untuk proses implementasi.

BAB V. PENGUJIAN DAN EVALUASI

Bab ini menjelaskan kemampuan perangkat lunak dengan melakukan pengujian fungsionalitas dan pengujian performa dalam beberapa skenario. Pengujian fungsionalitas merupakan pengujian jalannya aplikasi sesuai dengan perancangan sistem pada beberapa skenario yang telah ditentukan.

BAB VI. KESIMPULAN DAN SARAN

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan perangkat lunak ke depannya.

BAB II TINJAUAN PUSTAKA

Pada bab tinjauan pustaka akan dijelaskan beberapa hal mengenai teori yang berkaitan dengan sistem yang diimplementasikan. Hal ini ditujukan untuk memberikan gambaran secara umum terhadap sistem yang akan dibuat. Selain itu, hal tersebut berguna untuk menunjang pembuatan sistem sehingga kebutuhannya dapat diketahui.

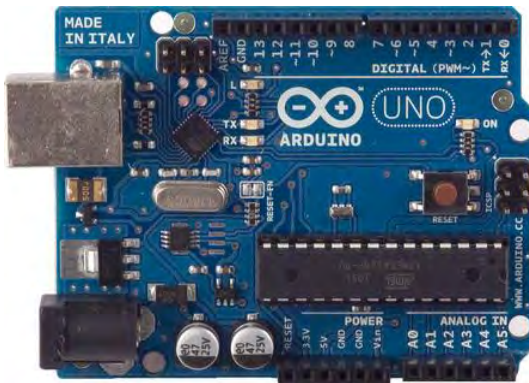
2.1 *Microcontroller* Arduino

Arduino merupakan sebuah *microcontroller* berbasis *single-board* yang bersifat *open-source*. Arduino dirancang untuk menggantikan tugas komputer sebagai alat untuk menerima inputan dari kondisi lingkungan sekitar. Arduino lebih ditujukan untuk mengendalikan perangkat keras dibandingkan komputer.

Board arduino dibangun menggunakan processor *Atmel AVR*. Arduino mendukung *input data digital* dan *analog* serta output data *digital*. Pada *board* arduino terdapat *boot-loader* arduino yang dapat di program dengan bahasa pemrograman C. Dalam sebuah *microcontroller* Arduino dapat pula ditanamkan berbagai macam *library* maupun metode selama kapasitas memori dari sebuah *microcontroller* mencakupi [1]. Arduino juga menggunakan *Integrated Development Environment (IDE)* berbasis *processing* dimana *processing* adalah bahasa *open-source* untuk menuliskan program ke komputer lainnya:

Jika terdapat sebuah *project* yang memerlukan beberapa komputer untuk berkomunikasi dengan Arduino, maka *processing* tersebut dapat digunakan, sehingga komputer-komputer tersebut dapat saling berkomunikasi dengan Arduino. Arduino memiliki kelebihan yaitu dapat dipasangkan dengan berbagai macam sensor, sebagai contoh yaitu sensor *accelerometer* berbasis *DFRobot* dan sensor *ultrasonic* berbasis *MaxSonar*. Untuk berkomunikasi

dengan komputer dapat menggunakan berbagai macam cara, antara lain menggunakan kabel serial, ethernet shield, GPRS/GSM module, wifi-shield, bluetooth dan lain-lain [1]. Arduino Uno merupakan board mikrokontroller dengan menggunakan ATmega328 sebagai basisnya. Di dalam Arduino Uno terdapat 14 pin sebagai *input/output digital* (6 pin diantaranya dapat digunakan sebagai *output PWM*), 6 pin *analog input*, 16 MHz *ceramics resonator*, sebuah koneksi USB, sebuah *socket power*, dan sebuah tombol *reset*. Arduino dapat dihidupkan dengan menggunakan kabel USB, sebuah *AC-to-DC adapter*, maupun baterai bertegangan 7-12 V. Program kapasitas memori Arduino Uno adalah 32 Kb dan 0.5 dari kapasistas memori digunakan untuk *bootloader*, serta memiliki SRAM sebesar 2 Kb dan 1 Kb sebagai EEPROM [2]. Perangkat *microcontroller* Arduino dapat dilihat pada Gambar 2.1.



Gambar 2.1 Perangkat *microcontroller* arduino

Aplikasi Arduino sudah terdapat *serial monitor* yang digunakan sebagai aplikasi uji coba untuk pengiriman dan menerima data dari komputer. Lampu LED pada TX/RX akan menyala ketika proses pengiriman dan penerimaan data pada *microcontroller* Arduino.

2.2 Sensor Anemometer

Fungsi anemometer yaitu sebagai perangkat atau alat yang digunakan untuk mengukur kecepatan angin. Dengan anemometer kita dapat memperkirakan cuaca pada hari tersebut. Selain itu, anemometer juga dapat difungsikan sebagai alat pendeteksi cuaca buruk seperti angin topan atau badai. Pada dasarnya anemometer adalah alat untuk mengukur kecepatan angin dalam fenomena terjadinya hembusan angin. Contohnya untuk mengukur aliran angin pada saluran, atau pengukuran arus terbatas, seperti angin atmosfer. Untuk menentukan kecepatan, anemometer umumnya mendeteksi perubahan di beberapa sifat fisik dari fluida pada alat mekanis dimasukkan ke dalam aliran.

Fungsi anemometer yang paling umum dapat kita temui pada anemometer mangkuk. Jenis ini merupakan alat yang paling umum digunakan karena merupakan anemometer yang paling sederhana. Anemometer ini terdiri dari beberapa mangkuk logam biasanya terdiri dari tiga pelekat pada ujung lengan horizontal yang terpasang pada ujung vertikal. Penangkapan dalam dalam mangkuk angin menyebabkan mereka berputar. Tindakan ini ternyata poros, yang terhubung ke perangkat yang memberikan kecepatan angin dalam mil per jam, kilometer per jam, atau knot.

2.3 *Weather Meters*

Anemometer berbentuk mangkuk ini mengukur kecepatan angin dengan cara memutuskan hubungan sebagai magnet yang bergerak melewati *switch*. Angin yang berkecepatan sebesar 2.4 km/jam menyebabkan *switch* menutup satu kali per detik.

2.4 Sensor Suhu Udara (DHT22)

Error! Reference source not found. merupakan gambar sensor suhu udara, sensor DHT22 adalah komponen dengan digital output mampu menghasilkan temperatur suhu udara yang akurat.



Gambar 2.2 Sensor DHT22

DHT22 memiliki keakuratan tinggi dan kemudahan perancangan jika dibandingkan dengan sensor suhu yang lain, DHT22 juga mempunyai kelebihan yaitu jarak transmisi hingga 20 meter serta penggunaan daya yang sangat rendah. Pada penelitian ini sensor suhu digunakan untuk mendeteksi suhu udara di sekitar node. *Datasheet* dari sensor DHT dapat dilihat pada tabel 2.1.

Tabel 2.1 *Datasheet* sensor DHT22

Nilai Output	°C (Celsius)
Presisi	+/- 0,5 °C
Range nilai	- 40 C – 80°C
Vcc input	3.3 – 6 V
Ukuran	2,5 x 1 x 0.8cm

2.5 Protokol ZigBee

ZigBee merupakan sebuah standar komunikasi *wireless* yang dikembangkan dari standar IEEE 802.15.4 [1]. ZigBee dikembangkan oleh aliansi perusahaan yang disebut dengan ZigBee Alliance. Lebih dari 300 perusahaan yang tergabung melakukan pengembangan dari standar komunikasi ZigBee. ZigBee memiliki beberapa keunggulan dan kelemahan sebagai standar komunikasi *wireless*. Keunggulan dari ZigBee adalah harga yang murah dan rendahnya konsumsi energi yang dibutuhkannya [3]. Keuntungan ini yang membuat ZigBee

Alliance selalu mencoba mengembangkan standar komunikasi ini. Akan tetapi ZigBee memiliki kelemahan dalam kecepatan transfer data. Kecepatan transfer data maksimal pada standar komunikasi ZigBee adalah 250 Kbps, oleh karena itu ZigBee jarang sekali digunakan untuk melakukan transfer data dalam jumlah yang besar. Berikut Tabel 2.1 perbandingan standar komunikasi ZigBee dengan standar komunikasi *wireless* lainnya.

Tabel 2.2 Perbandingan antara ZigBee, Bluetooth dan WiFi

Standar	<i>Bandwidth</i>	Konsumsi Daya	Kelebihan	Applikasi
Wi-fi	Lebih dari 54Mbps	400mA	Kecepatan transfer data	Internet
Bluetooth	1Mbps	40mA	mampu beroperasi antar <i>device</i>	komunikasi <i>handset</i> ,
ZigBee	250Kbps	30mA	Hemat energi dan murah	<i>Remote control, wireless sensor</i>

2.6 XBee Module

Xbee adalah sebuah merek yang dikeluarkan dari perusahaan Digi International untuk sebuah modul komunikasi radio. Perangkat Xbee dibuat oleh Digi International dengan memiliki 2 seri tipe yang berbeda. Setiap seri menggunakan standar komunikasi yang berbeda. Xbee seri 1 menggunakan standar komunikasi IEEE 802.15.4, sedangkan Xbee seri 2 dapat diatur menggunakan standar komunikasi IEEE 802.15.4 atau ZigBee. Berikut Tabel 2.2 perbandingan lengkap dari Xbee seri 1 dan seri 2.

Tabel 2.3 Perbandingan Xbee seri 1 dan Xbee seri 2

	Xbee Seri 1	Xbee Seri 2
Tegangan yang dikeluarkan	1 mW (0dbm)	2 mW (+3dbm)
kecepatan komunikasi data	250 Kbps	250 Kbps
Tegangan yang dibutuhkan	2.8 - 3.4 V	2.8 - 3.6 V
Frekuensi	ISM 2.4 GHz	ISM 2.4 GHz
temperatur alat	-40 to 85 C	-40 to 85 C
Bentuk jaringan	<i>Point to point, Star, Mesh (with DigiMesh firmware)</i>	<i>Point to point, Star, Mesh</i>
Kanal yang tersedia	<i>16 Direct Sequence Channels</i>	<i>16 Direct Sequence Channels</i>
Pilihan pengaturan	PAN ID, Channel & Source/Destination	PAN ID, Channel & Source/Destination

Dapat dilihat pada Tabel 2.2 bahwa Xbee seri 2 memiliki beberapa kelebihan dibandingkan dengan Xbee seri 1. Oleh karena itu sistem ini akan dibangun dengan menggunakan modul Xbee seri 2 yang memiliki banyak kelebihan dibandingkan dengan Xbee seri 1. Dalam hal ukuran Xbee seri 1 dan Xbee seri 2 memiliki ukuran yang sama. Perbedaannya hanya tulisan pada modul Xbee. S1 untuk Xbee seri 1 dan S2 Xbee seri 2. Gambar 2.3 menunjukkan gambar fisik Xbee seri 1 dan Xbee seri 2.



Gambar 2.3 Gambar A adalah Xbee seri 1 & Gambar B adalah Xbee seri 2

2.7 PHP

PHP (*Hypertext Preprocessor*) adalah sebuah bahasa pemrograman skrip yang banyak digunakan untuk pengembangan *web* dan dapat ditanamkan dalam bahasa *HyperText Markup Language* (HTML) [4]. PHP juga dapat digunakan sebagai bahasa pemrograman dengan tujuan umum. Saat ini telah digunakan di lebih dari 244 juta situs *web* dan 2,1 juta *server web*. Dibuat oleh Rasmus Lerdorf pada tahun 1995.

PHP merupakan bahasa pemrograman untuk aplikasi yang bekerja pada sisi *server*. Pada saat penjelajah *web* meminta dokumen PHP, *server web* langsung menggunakan modul PHP untuk mengolah dokumen tersebut. Jika pada dokumen terkandung fungsi yang mengakses basis data maka modul PHP menghubungkan *server* basis data yang bersangkutan. Dokumen yang berformat PHP dikembalikan [ke](#) *server web* dalam format HTML, sehingga kode PHP tidak tampak di sisi penjelajah *web*.

Kelebihan bahasa PHP dibandingkan dengan bahasa pemrograman lainnya antara lain:

- PHP merupakan bahasa skrip yang tidak melakukan sebuah kompilasi dalam penggunaannya.

- *Server web* yang mendukung PHP mudah ditemukan seperti Apache, IIS, Lighttpd, nginx, hingga Xitami dengan konfigurasi yang relatif mudah [4].
- Dari sisi pengembangan lebih mudah, karena banyaknya milis-milis dan tim pengembang yang siap membantu dalam pengembangan.
- Dari sisi pemahaman, PHP adalah bahasa pemrograman skrip yang paling mudah karena tersedia banyak referensi.

2.8 XBee Shield

Fungsi XBee *shield* ialah menghubungkan perangkat Arduino dan XBee *module*. XBee *shield* bisa berfungsi dengan baik pada XBee seri satu maupun dua. Gambar 2.3 adalah contoh XBee *shield*.



Gambar 2.4 XBee *shield*

2.9 Angin Puting Beliung

Angin puting beliung adalah angin kencang yang datang secara tiba-tiba, mempunyai pusat, bergerak melingkar seperti spiral hingga menyentuh permukaan bumi dan punah dalam waktu singkat (3-5 menit). Kecepatan angin rata-rata berkisar antara 30-40 knots atau 35 mile per hour sampai 46 mile per hour.

Puting beliung dapat terjadi dimana saja, di darat maupun di laut. Angin ini umumnya terjadi pada siang atau sore hari, lebih

sering terjadi pada musim pancaroba. Luas daerah yang terkena dampaknya sekitar 5-10 km, karena itu bersifat sangat lokal.

Angin puting beliung tidak dapat diprediksi secara spesifik. Namun dapat dirasakan tanda-tandanya sekitar 30 menit sampai 1 jam sebelumnya. Tanda-tandanya antara lain ialah suhu yang tinggi, panas, atau pengap. Lalu pepohonan disekitar kita bergoyang dengan cepat [5].

Pada sistem ini, rentang kecepatan angin yang diamati pada grafik yaitu kecepatan 0-19 MPH masuk ke dalam kategori angin rendah atau zona aman, lalu kecepatan angin 20-29 MPH merupakan zona angin sedang, dan kecepatan lebih besar dari 30 MPH merupakan angin puting beliung [6]. Suhu udara harian mempunyai rentang suhu sekitar 27 derajat Celcius sampai 34 derajat Celcius [7]

BAB III

PERANCANGAN PERANGKAT LUNAK DAN KERAS

Perancangan merupakan bagian terpenting dari pembuatan perangkat keras dan perangkat lunak yang berupa perencanaan – perencanaan secara teknis perangkat keras yang dibuat serta aplikasinya. Dalam bab perancangan perangkat lunak secara khusus akan menjelaskan perancangan sistem yang dibuat dalam Tugas Akhir. Berawal dari deskripsi umum hingga perancangan proses, alur dan implementasinya.

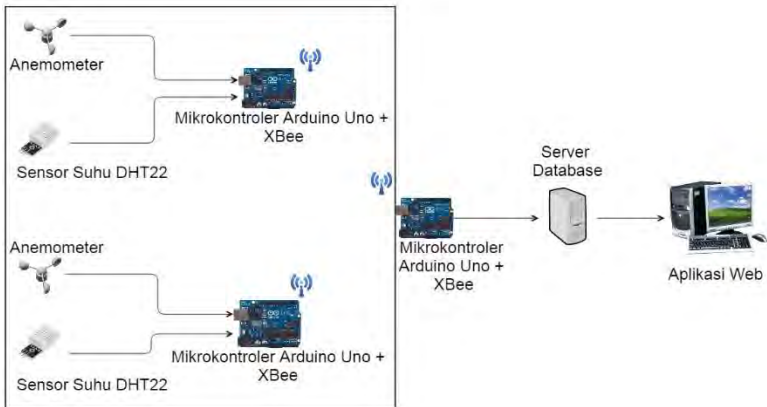
3.1 Deskripsi Sistem Secara Umum

Pada tugas akhir ini akan dibangun suatu sistem pemantauan cuaca (angin) dengan menggunakan sensor DHT22 yang digunakan sebagai pengukur suhu dan sensor anemometer. Sistem pemantauan cuaca angin ini dirancang untuk mengetahui kondisi kecepatan angin dan suhu udara. Data yang dipantau oleh sistem ini ialah besar kecepatan angin dan besarnya suhu.

Pada tugas akhir dibutuhkan dua buah sensor untuk memantau cuaca, yaitu sensor anemometer dan sensor DHT22 yang dihubungkan dengan menggunakan *microcontroller* Arduino sebagai pengolah data yang kemudian akan dikirimkan dengan menggunakan XBee. Untuk pengiriman data penulis menggunakan sebuah perangkat Xbee. Pengiriman data dari XBee menuju ke pos pemantau cuaca dilakukan secara terus – menerus. Pada pos pemantau cuaca terdapat sebuah komputer *server* yang digunakan untuk mengolah data menjadi sebuah hasil yang interaktif. Sistem ini juga dirancang agar bisa mendeteksi potensi dini angin puting beliung.

3.2 Arsitektur Umum Sistem

Rancangan arsitektur dari sistem yang akan dibuat terlihat pada Gambar 3.1 yang terdapat beberapa komponen dan sensor utama.



Gambar 3.1 Arsitektur Sistem Secara Umum

a. Sensor Anemometer

Sensor anemometer berfungsi untuk mendapatkan nilai kecepatan angin dalam satuan *mile per hour*. Hanya terdapat satu nilai pada sensor ini. setiap angin bergerak maka sensor ini akan mengikuti dan mendapatkan nilai yang berupa data, selanjutnya dari setiap data tersebut akan diteruskan menuju *microcontroller*.

b. Sensor DHT22

Sensor DHT22 adalah sensor pengukur suhu. Berfungsi untuk mendapatkan nilai suhu udara dalam satuan Celcius. Data dari sensor ini akan diteruskan menuju *microcontroller*.

c. XBee Module

XBee dalam sistem pemantauan ini berfungsi sebagai pengirim data antara *microcontroller* arduino dan *server*. Jarak maksimal yang bisa diakomodasi oleh modul XBee adalah 30 meter *indoor* dan 100 meter *outdoor*.

d. *Microcontroller Arduino*





Microcontroller Arduino berguna untuk memproses nilai yang didapatkan oleh sensor anemometer dan sensor DHT22. Pada *microcontroller* data dari sensor anemometer dan sensor DHT22 akan dikirimkan ke server dengan menggunakan perangkat XBee modul. Perangkat komunikasi penulis menggunakan XBee untuk dapat mengirimkan data dari *microcontroller*.

e. *XBee shield*

Xbee *shield* berfungsi sebagai penghubung antara *microcontroller* arduino dan XBee module. Dengan menggunakan XBee *shield*, kita tidak perlu membuat rangkaian tambahan untuk menghubungkan antara XBee module dan *microcontroller*.

3.3 Perancangan Basis Data

Perancangan basis data menggambarkan struktur tabel yang akan digunakan dalam penyimpanan kebutuhan data sistem pada basis data. Pada Gambar 3.2 menggambarkan *Database Model* dari basis data yang digunakan dalam sistem ini. Terdapat 3 tabel dengan masing-masing atribut yang digunakan untuk menyimpan kebutuhan sistem.

datanode	
	id_pengirim: CHAR(10)
	waktu: TIMESTAMP
	kecepatan_angin: FLOAT
	suhu_udara: FLOAT

Gambar 3.2 Perancangan Basis Data

3.4 Perancangan Perangkat Keras

Pada sistem ini, perangkat keras yang digunakan sebagai berikut:

1. 3 buah *microcontroller* Arduino.
2. 2 buah sensor anemometer.
3. 2 buah sensor suhu DHT22.
4. 3 buah alat komunikasi XBee Module.
5. 1 buah soket baterai.
6. 1 buah baterai 9 V.
7. 1 buah power bank.
8. 1 buah komputer Acer tipe intel(R) Core(TM) i3-3240 CPU @ 3.40 GHz memory 4096 MB RAM.

3.4.1 Perancangan Perangkat Keras ZigBee Koordinator

Perancangan perangkat keras ZigBee koordinator menggunakan satu ZigBee modul seri 2, satu Arduino Xbee *shield*, dan satu Arduino UNO. Pemasangan setiap perangkat dengan menempelkan setiap perangkat seperti pada Gambar 3.3.



Gambar 3.3 Perancangan Perangkat ZigBee Koordinator

3.4.2 Perancangan Perangkat Keras ZigBee *End Device*

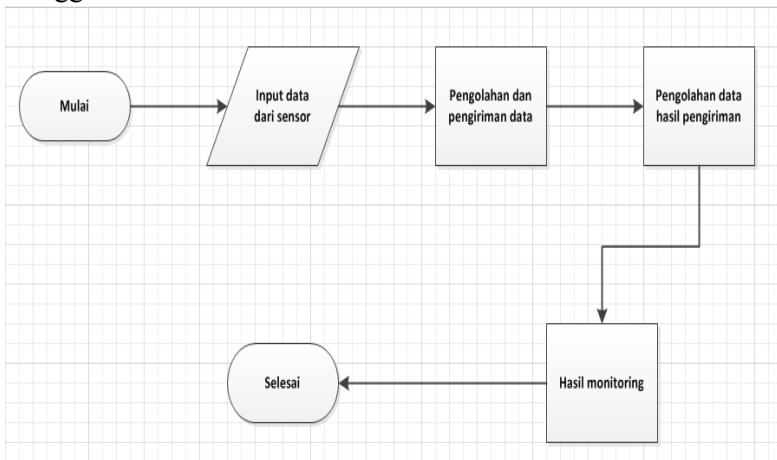
Perancangan perangkat keras ZigBee koordinator menggunakan satu ZigBee modul seri 2, satu Arduino Xbee *shield*, satu Arduino *ethernet shield*, dan satu Arduino UNO. Pemasangan setiap perangkat dengan memasang setiap perangkat seperti pada Gambar 3.4.



Gambar 3.4 Perancangan Perangkat ZigBee *End Device*

3.5 Diagram Alir Aplikasi

Pada diagram alir sistem pemantauan cuaca menggambarkan urutan fungsionalitas sistem secara keseluruhan. Diagram alir sistem dapat dilihat pada gambar 3.5. Sistem diawali dengan membaca input data dari sensor anemometer dan sensor DHT22. Kemudian data yang dihasilkan oleh sensor akan diolah sehingga menghasilkan data. Data tersebut akan dikirimkan menggunakan XBee module.



Gambar 3.5 Perancangan Diagram Alir Sistem

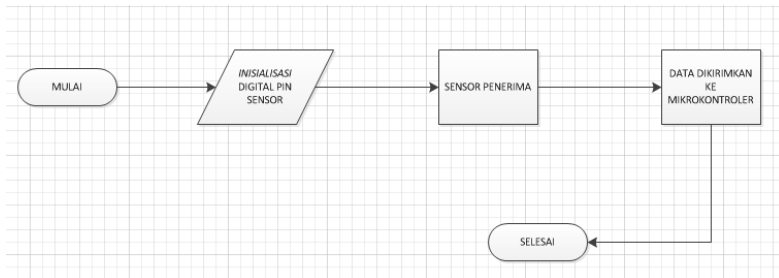
3.6 Diagram Alir Aplikasi Sistem

Alur setiap proses yang terdapat pada aplikasi digambarkan pada diagram alir, untuk memudahkan pemahaman secara garis besar proses yang ada pada sistem.

3.6.1 Diagram Alir Mendeteksi Nilai Suhu Udara

Nilai suhu udara diambil dari sensor suhu udara DHT22. Sensor ini membutuhkan satu input *digital* untuk dapat beroperasi pada *microcontroller* Arduino. Pembacaan nilai sensor ini terjadi

secara *realtime* atau secara terus-menerus. Diagram alir mendeteksi nilai suhu udara dapat dilihat pada gambar 3.6.

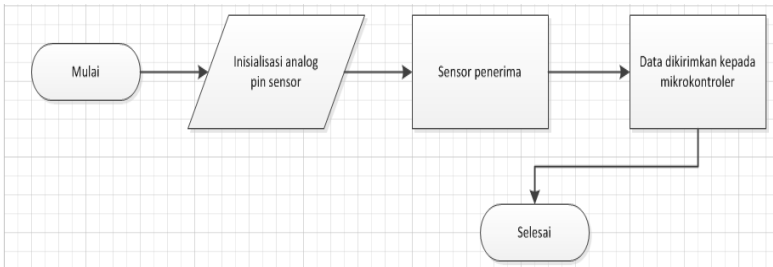


Gambar 3.6 Diagram Alir Mendeteksi Nilai Suhu Udara

Proses yang terlihat pada gambar 3.6 merupakan proses pengambilan nilai suhu udara. Penulis melakukan proses pengambilan nilai suhu udara ini secara berulang.

3.6.2 Diagram Alir Mendeteksi Nilai Kecepatan Angin

Pada sensor anemometer yang berfungsi untuk mengukur kecepatan angin dengan mengitung besarnya gerakan mangkuk pada anemometer, dari jumlah putaran dalam satu detik maka akan diketahui kecepatan anginnya. Di dalam anemometer terdapat alat pencacah yang akan menghitung kecepatan angin. Didapatkan data kecepatan angin yang telah diterima dan selanjutnya *microcontroller* Arduino yang bertugas untuk mengolah data tersebut. Diagram alir mendeteksi nilai kecepatan angin dapat dilihat pada gambar 3.7.

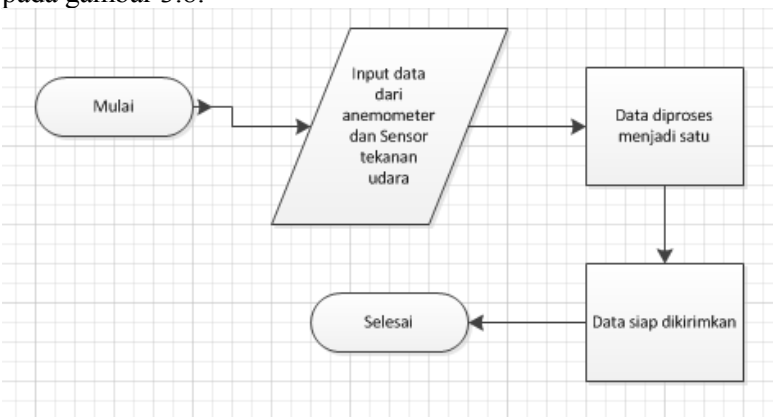


Gambar 3.7 Diagram Alir Mendeteksi Nilai Kecepatan Angin

Pada gambar 3.7 proses mendapatkan nilai kecepatan angin ini hanya dilakukan satu kali. Penulis melakukan pengambilan kecepatan angin ini secara berulang.

3.6.3 Diagram Alir *Microcontroller*

Hasil data *output* dari sensor anemometer dan sensor suhu yang selanjutnya akan diproses pada *microcontroller* Arduino. *Microcontroller* ini akan menggabungkan keseluruhan data yang diperoleh sehingga didapatkan kesatuan data yang nanti akan dikirimkan oleh XBee. Diagram alir *microcontroller* dapat dilihat pada gambar 3.8.

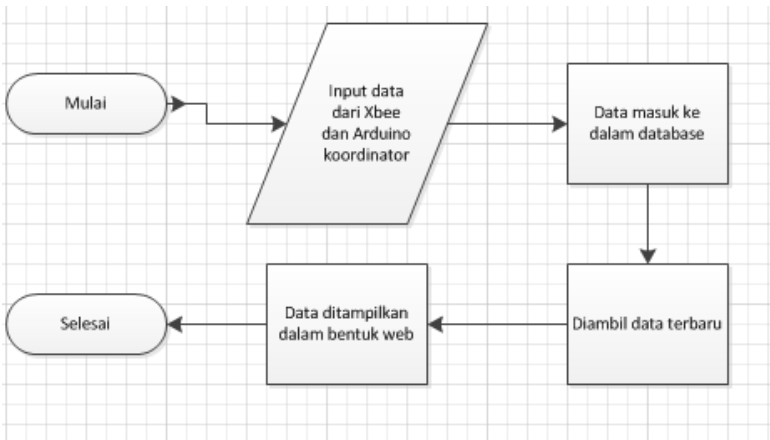


Gambar 3.8 Diagram Alir *Microcontroller*

3.6.4 Diagram Alir Penerimaan Data Pada Komputer

Agar komputer dapat melakukan penerimaan data dengan *microcontroller*, maka komputer harus terhubung dengan perangkat ZigBee koordinator. Komputer telah mendapatkan domain untuk dapat melakukan penerimaan data dari XBee.

Data tersebut selanjutnya akan masuk kedalam database dan akan ditampilkan data dalam bentuk grafik sehingga dapat memudahkan pengguna untuk *monitoring* dari nilai kecepatan angin dan suhu udara. Diagram alir inialisasi koneksi pada aplikasi *web* dapat dilihat pada gambar 3.9.



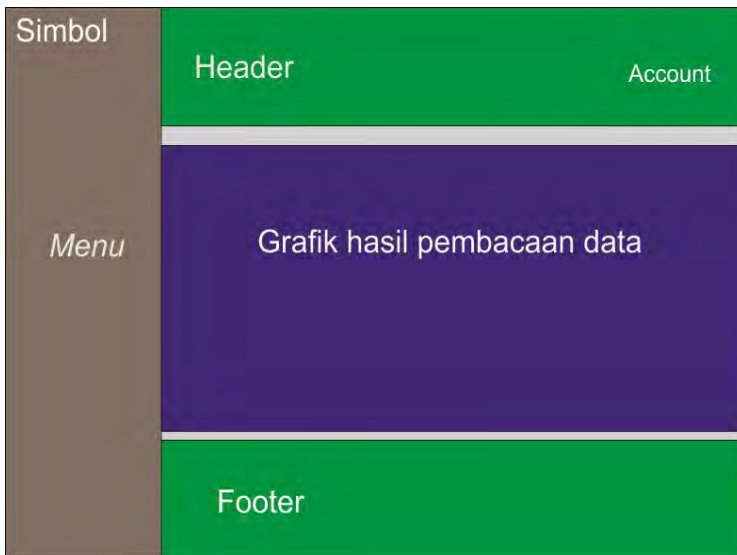
Gambar 3.9 Diagram Alir Penerimaan Data Pada Komputer

3.6.5 Rancangan Antar Muka Aplikasi

Sistem berjalan secara otomatis dan user hanya menggunakan web aplikasi ini sebagai alat bantu pemantauan kecepatan angin dan suhu udara. Rancangan antar muka aplikasi dibuat dalam bentuk aplikasi *web*.

Pada gambar 3.10 data ditampilkan dalam bentuk grafik. Pada grafik ini hasil dari pembacaan data ditampilkan secara *realtime* sehingga mempermudah *user*. *User* dapat memantau kecepatan angin dan suhu udara. Node akan bertambah secara otomatis ketika ada node yang ditambahkan.

Terdapat satu halaman untuk menampilkan semua data secara bersamaan, yaitu halaman index. Pada bagian samping aplikasi web, terdapat menu yang berisi node aktif.



Gambar 3.10 Rancang Bangun *Web* Aplikasi

BAB IV IMPLEMENTASI

Pada bab ini dibahas implementasi dari rancangan perangkat lunak yang didasari pada rancangan pada bab 3. Implementasi meliputi implementasi pada perangkat ZigBee dan aplikasi *web*.

4.1 Lingkungan Implementasi

Untuk mengimplementasikan perangkat lunak ini digunakan beberapa perangkat pendukung sebagai berikut.

4.1.1 Lingkungan Implementasi Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan sistem *monitoring* kecepatan angin dan besar suhu udara, *microcontroller* Arduino, XBee *shied*, modul XBee seri 2. Adapun spesifikasi yang digunakan adalah sebagai berikut:

- 1 buah komputer desktop lenovo tipe intel(R) Core(TM) i3-3240 CPU @ 3.40 GHz.
- 3 buah perangkat Arduino UNO, Atmega328P dengan 32kb *flash* memori.
- 3 buah XBee *shield*.
- 3 buah modul XBee seri 2 dari Digi International.
- 2 buah sensor anemometer.
- 2 buah sensor suhu udara DHT22.
- 2 buah baterai Panasonic 9 V.

4.1.2 Lingkungan Implementasi Perangkat Lunak

Perangkat lunak yang digunakan dalam implementasi sistem *monitoring* kecepatan angin dan suhu udara, yaitu:

- Microsoft Windows 7 x86 sebagai sistem operasi.
- Python 2.7 sebagai implementasi aplikasi *server*.

- Mozilla Firefox sebagai *browser* dalam tahap uji coba aplikasi *server*.
- Arduino Development kit sebagai IDE untuk mengimplementasikan IDE untuk mengimplementasikan aplikasi Arduino.
- XAMPP *version 3.2.1* sebagai server yang terdiri atas program:
 - Apache HTTP Server sebagai aplikasi *web server* untuk menghasilkan halaman web yang benar kepada *user* berdasarkan kode PHP.
 - MySQL sebagai aplikasi *database server* untuk mengolah dan menyimpan data.
- X-CTU sebagai aplikasi untuk memasang *firmware* pada modul XBee.

4.2 Implementasi Perangkat Keras ZigBee Koordinator

Pada implementasi perangkat keras ZigBee koordinator dibutuhkan 4 perangkat keras yaitu:

- 1 buah Arduino UNO.
- 1 buah Arduino Xbee *shield*.
- 1 buah modul Xbee seri 2.

Semua perangkat tersebut kemudian disusun sesuai dengan Gambar 4.1 untuk bisa menjadi perangkat ZigBee koordinator.



Gambar 4.1 Perangkat Keras yang Dibutuhkan ZigBee Koordinator

Perangkat keras *microcontroller* Arduino bertugas sebagai pusat dari data proses terhadap setiap data yang diterima. XBee *shield* bertugas sebagai alat bantu penghubung antara modul XBee dengan *microcontroller* Arduino. Apabila semua perangkat sudah terpasang, maka perangkat keras ZigBee koordinator akan seperti Gambar 4.2.

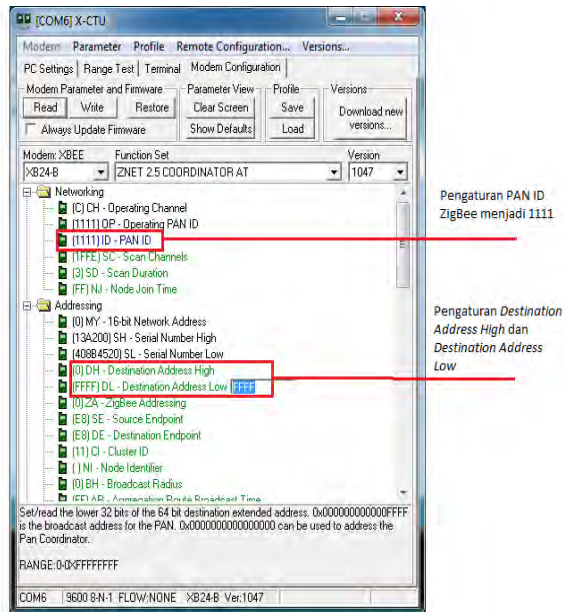


Gambar 4.2 ZigBee Koordinator Setelah di Rakit

4.3 Implementasi *Firmware* ZigBee Koordinator

Sebelum melakukan implementasi perangkat lunak terhadap ZigBee koordinator terlebih dahulu dilakukan pemasangan *firmware* terhadap perangkat ZigBee. Pemasangan *firmware* pada ZigBee menggunakan aplikasi X-CTU yang dikeluarkan oleh perusahaan Digi International.

Pemasangan *firmware* pada ZigBee koordinator akan menggunakan tipe *firmware* Znet 2.5 COORIDINATOR AT. Setelah menentukan tipe *firmware*, ZigBee koordinator perlu diberi pengaturan PAN ID (*Personal Area Network Identification*), *destination address high* dan *destination address low*. Pengaturan ZigBee koordinator dapat dilihat pada Gambar 4.3.



Gambar 4.3 Pemasangan *Firmware* ZigBee Koordinator Dengan Menggunakan X-CTU

Pengaturan pada Gambar 4.3 terlihat bahwa PAN ID ZigBee koordinator diubah menjadi 1111. Pengaturan *destination address high* menjadi 0 karena ini adalah perangkat ZigBee koordinator dan *destination address low* menjadi FFFF untuk pesan *broadcast* kepada setiap perangkat ZigBee *end device*.

4.4 Implementasi Perangkat Lunak ZigBee Koordinator

Implementasi perangkat lunak pada ZigBee koordinator membutuhkan sebuah array of *character* untuk menampung data yang sudah dibaca. ZigBee koordinator membaca data yang dikirim oleh ZigBee *end device* per *byte*, sehingga array of *character* dibutuhkan pada ZigBee koordinator.

Setelah data yang ditampung sudah mencukupi, maka proses selanjutnya ialah mencetak data tersebut. Implementasi fungsi ini dapat dilihat pada Gambar 4.4.

```

input : void
output : void
-----
function zigBeeKoor()
if client available
while (true)
set inChar <- read char from client

if inChar = "<"
start          <- start array
else if inChar = ">"
stop          <- stop array
else
              inData[index] = inChar <- memasukkan char pada
                                   array
send inData[]
end function

```

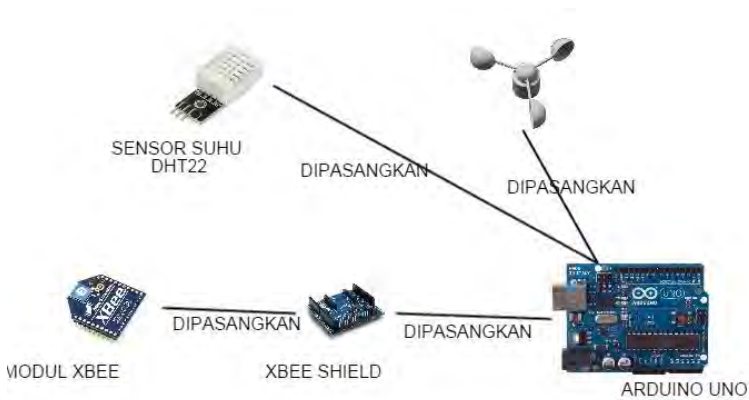
Gambar 4.4 Implementasi Perangkat Lunak ZigBee Koordinator

4.5 Implementasi Perangkat Keras ZigBee *End Device*

Pada implementasi perangkat keras ZigBee *end device*, perangkat yang dibutuhkan tidak jauh berbeda dengan perangkat ZigBee koordinator. Berikut perangkat ZigBee *end device* yang dibutuhkan yaitu:

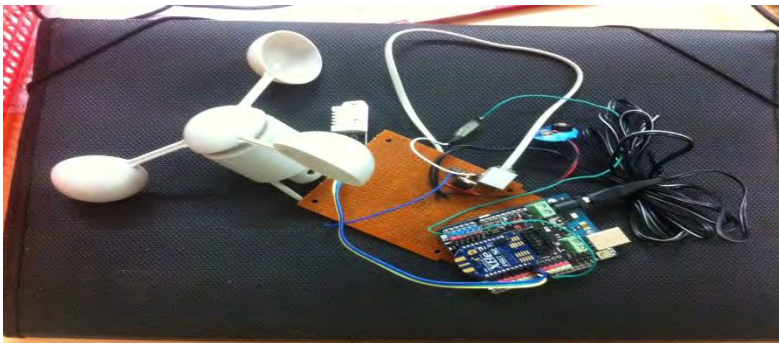
- Satu buah Arduino Uno.
- Satu buah Arduino XBee *shield*.
- Satu buah anemometer.
- Satu buah sensor suhu udara MPX2200AP.

Semua perangkat tersebut kemudian disusun serupa dengan ZigBee koordinator. Sesuai dengan Gambar 4.5 untuk bisa menjadi perangkat ZigBee *end device*.



Gambar 4.5 Perangkat ZigBee *End Device*

Gambar 4.6 akan terlihat ZigBee *end device* yang terdapat perangkat sensor yang dipasang pada ZigBee *end device*. Sensor yang dipasang adalah sensor suhu DHT22 dan anemometer. Sensor suhu digunakan sebagai pendeteksi suhu udara. Sedangkan anemometer digunakan sebagai pengukur kecepatan angin.

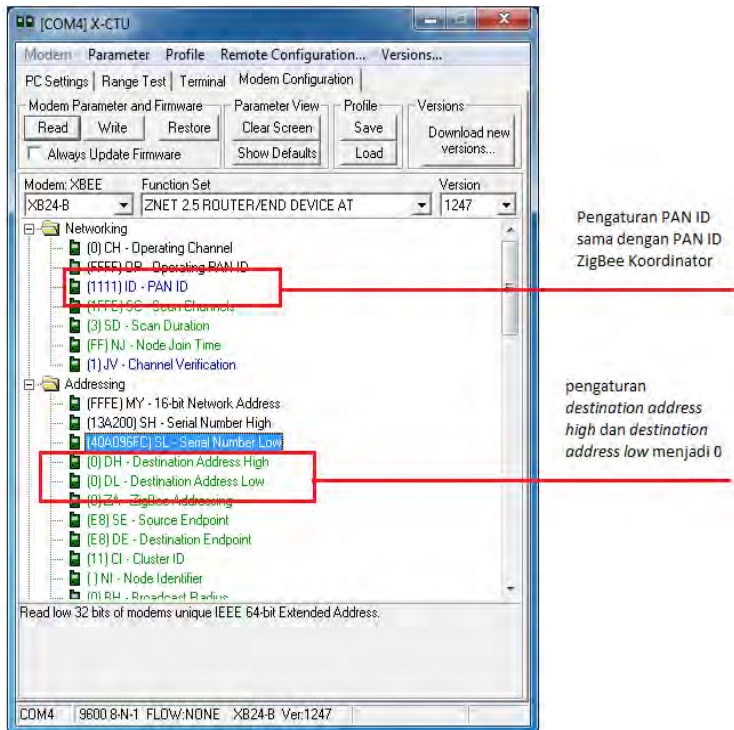


Gambar 4.6 ZigBee *End Device* Setelah Dirakit

4.6 Implementasi *Firmware ZigBee End device*

Setelah pemasangan semua perangkat keras ZigBee *end device*, perangkat ZigBee *end device* harus dipasang *firmware*

ZigBee tipe *end device*. Pemasangan *firmware* menggunakan aplikasi yang sama ketika pemasangan *firmware* pada ZigBee koordinator. Pengaturan *firmware* pada ZigBee *end device* ditunjukkan pada Gambar 4.7.



Gambar 4.7 Pemasangan *Firmware* pada ZigBee *End Device*

Pada Gambar 4.8 terlihat bahwa PAN ID ZigBee *end device* diubah serupa dengan ZigBee koordinator. Menyamakan nilai PAN ID pada ZigBee *end device* dengan ZigBee koordinator dimaksudkan bahwa ZigBee *end device* akan mengikuti jaringan ZigBee dengan koordinator yang telah dibuat sebelumnya.

4.7 Implementasi Perangkat Lunak ZigBee *End Device*

Implementasi perangkat lunak pada ZigBee *end device* berada pada fungsi *loop* mikrokontroler Arduino. Fungsi *loop* akan berisi data array berisi *byte* data yang sudah ditampung sebelumnya. Array berisi data suhu udara dan kecepatan angin. Implementasi fungsi tersebut ditunjukkan pada Gambar 4.8

```
input : void
output : void
-----
wind speed = pin 8      <- inisialisasi pin
temp = pin 7           <- inisialisasi pin
function loop()
    set dataPacket(id, wind speed, temperature)
    send dataPacket
end function
```

Gambar 4.8 Implementasi Perangkat Lunak ZigBee *End Device*

4.8 Implementasi Pada *Server*

Implementasi perangkat lunak pada ZigBee *end device* berada pada fungsi *loop* mikrokontroler Arduino. Fungsi *loop* akan berisi data array berisi *byte* data yang sudah ditampung sebelumnya. Array berisi data suhu udara dan kecepatan angin. Implementasi fungsi tersebut ditunjukkan pada Gambar 4.8.

4.9 Implementasi Memasukkan Data ke Basis Data

Implementasi pertama yaitu memasukkan data yang telah diterima oleh ZigBee koordinator kepada database. Proses ini dapat dilihat pada gambar 4.9.

```
input : void
output : void
-----
function getData()
openConnection()
    if serial available
        while(true)
            arduino.read() <- read data
            insert into database <- insert data
        closeConnection
```



```

        else
            send failed
    end function

```

Gambar 4.9 Implementasi Memasukkan Data ke Basis Data

4.10 Implementasi Menampilkan Data pada *Web*

Implementasi selanjutnya adalah penampilan data dari database oleh aplikasi *web*. Pada implementasi ini, data yang sudah diterima oleh database akan ditampilkan ke dalam aplikasi *web* dengan menggunakan *script* PHP. Implementasi fungsi tersebut dapat dilihat pada Gambar 4.10.

```

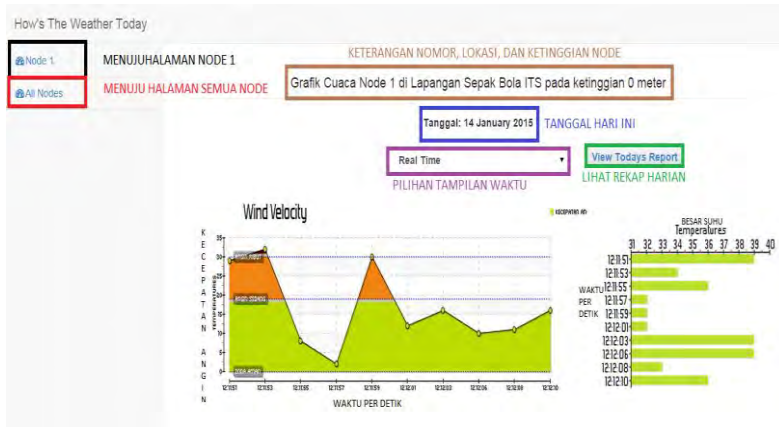
input  : id, wind speed, air pressure, time
output : void
-----
function showData()
    open basis data
    select from datanode
    get id, wind speed, air pressure, time
    show chart
    close basis data
end function

```

Gambar 4.10 Implementasi Menampilkan *Chart* pada *Web*

4.11 Implementasi Tampilan Grafik pada *Web*

Implementasi tampilan berhubungan langsung dengan implementasi dari antarmuka aplikasi *web*. Implementasi antarmuka terdiri dari tiga antarmuka, yaitu antarmuka untuk menampilkan grafik masing-masing node (dua antarmuka), dan antarmuka untuk menampilkan gabungan dari semua node.



Gambar 4.11 Antarmuka node 1

Pada Gambar 4.11 diperlihatkan tampilan antarmuka untuk menampilkan grafik dari node 1. Tampilan ini dinamis, dalam artian akan bertambah apabila node sensor *end device* bertambah. Dapat dilihat pada Gambar 4.11, gambar yang berjudul *wind velocity* adalah grafik yang menunjukkan besar kecepatan angin, sedangkan grafik yang berjudul *temperatures* adalah grafik yang menunjukkan suhu udara.

Pada Gambar 4.12 tampilan antarmuka untuk menampilkan rekap data kecepatan angin dan suhu udara pada hari tersebut. Data yang direkap antara lain ialah rata-rata kecepatan angin dan suhu udara, lalu nilai minimum dan maksimum kecepatan angin dan suhu udara. Nilai ini di rekap setiap harinya. Akan disediakan menu untuk mengunduh data rekap harian ini.

How's The Weather Today

All Nodes

Sistem Pemantauan Arduino

Rata-rata Kecepatan Angin	17.62 MPH
Kecepatan Angin Maksimum	31 MPH pada pukul 09:24:28
Kecepatan Angin Minimum	3 MPH pada pukul 09:24:28
Rata-rata Suhu	36.02 derajat Celcius
Suhu Maksimum	40 derajat Celcius pada pukul 09:24:28
Suhu Minimum	31 derajat Celcius pada pukul 09:24:28

Export to Excel

Gambar 4.12 Antarmuka rekap nilai harian

BAB V

PENGUJIAN DAN EVALUASI

Bab ini akan membahas uji coba dan evaluasi dari sistem yang dibuat. Sistem akan diuji coba fungsionalitas dan performa dengan menjalankan skenario yang sudah ditentukan. Uji coba dilakukan untuk mengetahui hasil dari sistem ini sehingga dapat menjawab rumusan masalah pada Bab 1.

5.1 Lingkungan Uji Coba

Uji coba fungsionalitas dilakukan untuk mengetahui fungsi dasar masing-masing komponen berjalan sebagaimana mestinya. Untuk itu dilakukan uji coba dengan menjalankan masing-masing fungsinya sehingga mengeluarkan hasil yang sesuai. Pada sistem terbagi menjadi 2 bagian yaitu bagian perangkat ZigBee dengan Arduino dan bagian *server*. Bagian Arduino terdiri dari ZigBee koordinator dan ZigBee *end device*. Sedangkan bagian server adalah laptop yang bertugas untuk mengolah dan menampilkan data. Data yang diukur mempunyai satuan Celcius untuk suhu udara, dan mile per jam untuk kecepatan angin.

Uji coba fungsionalitas dilakukan di lantai 4 Teknik Informatika ITS. Sedangkan uji coba performa sistem dilakukan di lantai 4 Teknik Informatika ITS dan lapangan sepak bola stadion ITS. Pada Gambar 5.1 akan terlihat suasana lingkungan uji coba perangkat ZigBee *end device*.



Gambar 5.1 Suasana uji coba ZigBee *end device* di lantai 4 Teknik Informatika ITS

Gambar 5.2 merupakan suasana uji coba pada ZigBee koordinator. ZigBee koordinator disambungkan dengan laptop melalui kabel *serial*.



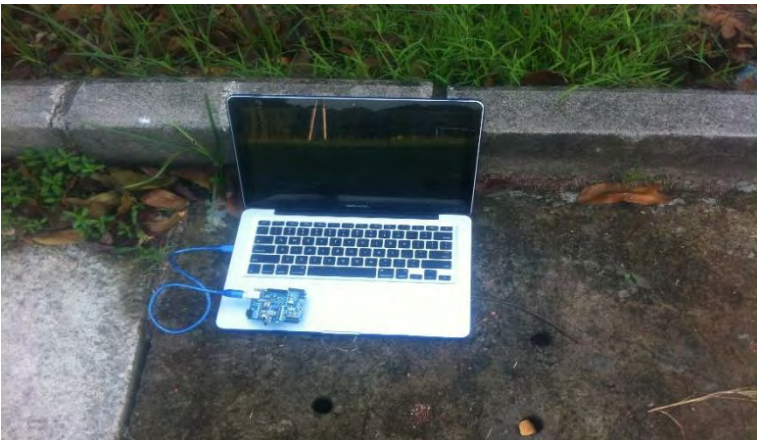
Gambar 5.2 Suasana uji coba pada ZigBee koordinator di lantai 4 Teknik Informatika ITS

Pada Gambar 5.3 akan terlihat suasana lingkungan uji coba perangkat ZigBee *end device* pada lapangan sepak bola.



Gambar 5.3 Suasana uji coba pada ZigBee *end device* di lapangan sepak bola ITS

Gambar 5.4 merupakan suasana uji coba pada ZigBee koordinator.



Gambar 5.4 Suasana uji coba pada ZigBee koordinator di lapangan sepak bola ITS

Pada subbab ini, dijelaskan mengenai gambaran lingkungan yang digunakan sebagai uji coba sistem *monitoring* suhu. Uji coba

sensor dilakukan di luar ruangan. Pengiriman data dari perangkat ZigBee koordinator kepada *server* akan melalui kabel serial. Lingkungan uji coba memiliki spesifikasi sebagai berikut:

- Perangkat ZigBee koordinator (*microcontroller* Arduino, modul XBee seri 2, XBee *shield*).
- Perangkat ZigBee *end device* (*microcontroller* Arduino, modul XBee seri 2, XBee *shield*, anemometer, sensor suhu DHT22).
- *Powerbank* 2600mah Oren.
- 2 baterai 9V.
- Laptop Macbook Pro, Intel i5 2.5GHz dual-core GHz dan 2 GB memoriDDR3 dengan Windows 7 Ultimate.

5.1.1 Uji Coba Mendapatkan Data Kecepatan Angin dan Suhu Udara

Pada sistem terdapat fungsi untuk mendapatkan data kecepatan angin dan suhu udara secara *real time*. ZigBee *end device* akan mengirimkan data dimana nantinya ZigBee koordinator akan membaca data. Data yang telah diterima oleh ZigBee koordinator akan dikirimkan kepada *server* untuk disimpan di dalam basis data. Untuk skenario uji coba mendapatkan data kecepatan angin dan suhu udara dapat dilihat pada Tabel 5.1.

Tabel 5.1 Skenario uji coba mendapatkan data kecepatan angin dan suhu udara

Nama	Uji coba mendapatkan data kecepatan angin dan suhu udara
Tujuan	Mendapatkan data kecepatan angin dan suhu udara
Skenario	ZigBee <i>end device</i> diletakkan pada jarak 10 meter dari ZigBee koordinator
Hasil Uji Coba	Data kecepatan angin dan suhu udara berhasil didapatkan dan ditampilkan oleh aplikasi <i>web</i>

Uji coba untuk mendapatkan data kecepatan angin dan suhu udara adalah dengan melakukan pengujian di dekat kipas angin.

Gambar 5.5 adalah gambar antarmuka grafik kecepatan angin dan suhu udara pada node 1. Rentang garis bagian bawah adalah besar kecepatan angin, sedangkan rentang bagian atas adalah besar suhu udara. Dengan sumbu-x pada grafik menunjukkan waktu, dan sumbu-y menunjukkan besar kecepatan angin dan suhu udara.



Gambar 5.5 Uji coba mendapatkan data kecepatan angin dan suhu udara

5.1.2 Uji Coba Menampilkan Peringatan dan Alarm Potensi Puting Beliung

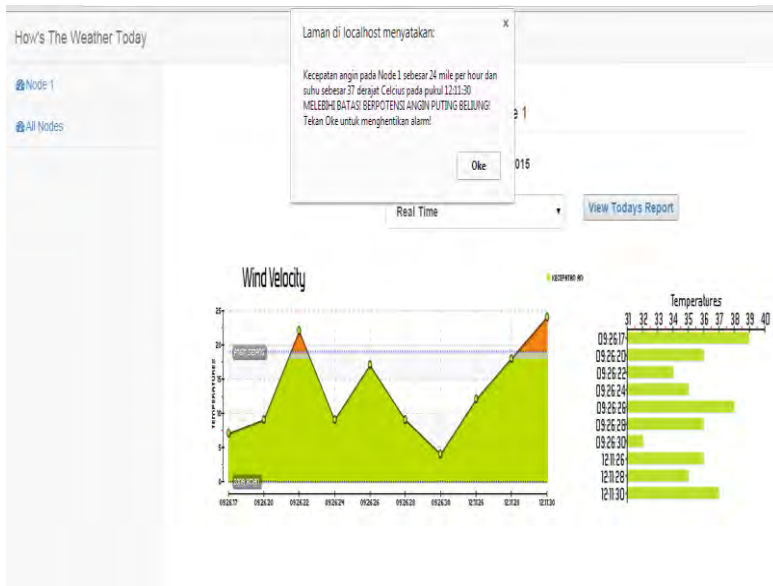
Uji coba menampilkan peringatan dan alarm potensi puting beliung ini bertujuan apakah sistem dapat menampilkan peringatan dan juga alarm jika kecepatan angin dan suhu menunjukkan potensi terjadinya angin puting beliung. Pada sistem ini diberi rule, jika kecepatan angin melebihi 30 MPH, dan suhu melebihi rata-rata, (34 derajat Celcius), maka alarm dan kotak

peringatan akan muncul. Skenario uji coba ini dapat dilihat pada Tabel 5.2.

Tabel 5.2 Skenario uji coba menampilkan peringatan dan alarm potensi puting beliung

Nama	Uji coba menampilkan peringatan dan alarm potensi puting beliung
Tujuan	Menampilkan alert dan alarm jika terjadi potensi angin puting beliung
Skenario	Memasukkan data sesuai <i>rule</i> agar menyerupai potensi angin puting beliung
Hasil Uji Coba	Alarm dan kotak peringatan muncul dengan baik

Gambar 5.6 menunjukkan kotak dan alarm peringatan adanya potensi angin puting beliung.



Gambar 5.6 Grafik antarmuka dengan kotak peringatan dan alarm

5.2 Uji Coba Performa Sistem

Uji coba performa sistem dilakukan untuk mengetahui seberapa baik sistem yang dibangun. Dalam uji coba performa sistem yang dilakukan adalah menghitung jumlah kesalahan pengiriman data yang terjadi pada sistem, dengan jarak yang berbeda-beda. Dengan menghitung jumlah kesalahan data yang terjadi pada sistem, bisa diketahui jarak yang ideal untuk sistem ini bekerja.

5.2.1 Uji Coba Sistem pada Lantai 4 Teknik Informatika ITS

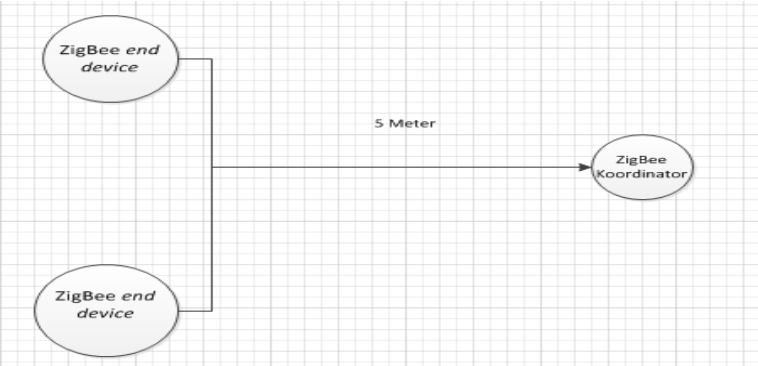
Uji coba sistem pada Lantai 4 Teknik Informatika ini ZigBee koordinator dan ZigBee *end device* diberi jarak sebesar 5 meter, 10 meter, dan 15 meter. Jarak ini dipilih karena dianggap ideal untuk komunikasi XBee. Skenario uji coba ini dapat dilihat pada Tabel 5.3.

Tabel 5.3 Skenario uji coba pada lantai 4 Teknik Informatika

Nama	Uji coba pada lantai 4 Teknik Informatika ITS
Tujuan	Membandingkan data yang diterima pada berbagai jarak
Skenario	ZigBee <i>end device</i> dan ZigBee koordinator ditempatkan pada berbagai jarak
Hasil Uji Coba	Data berhasil didapat dan dibandingkan

5.2.1.1 Uji Coba pada Jarak 5 Meter

Gambar 5.7 adalah visualisasi jarak antara ZigBee *end device* dan ZigBee koordinator.



Gambar 5.7 Visualisasi jarak ZigBee end device dan koordinator

Uji coba pada jarak ini, data yang diterima hanya mengalami sedikit kesalahan. Dapat dilihat pada Tabel 5.2, pada data ke 30 dari nomor urut ZigBee end device hilang. Data yang mempunyai kesalahan merupakan data mempunyai nomor urut ZigBee end device yang tidak berurutan. Maka, jumlah *packet loss* sistem adalah 1. Tabel 5.4 adalah hasil uji coba jarak 5 meter.

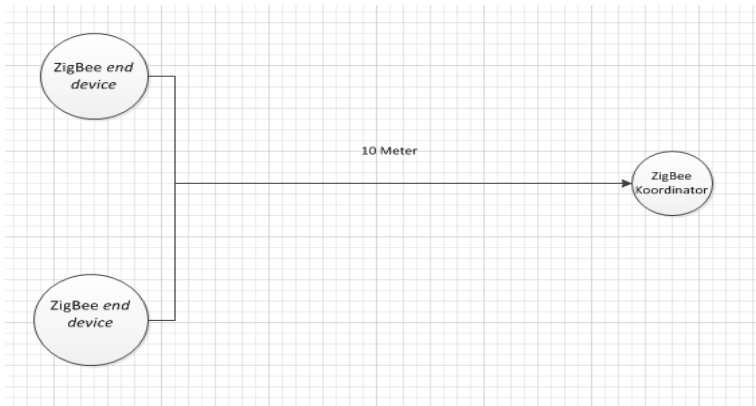
Tabel 5.4 Data uji coba jarak 5 meter

WAKTU	ID NODE	KECEPATAN ANGIN (MPH)	SUHU UDARA (CELCIUS)	NO URUT END DEVICE	NO URUT KOORDI- NATOR
16:44:58	1	16	27	24	1
16:45:03	1	15	27	25	2
16:45:05	1	17	27	26	3
16:45:07	1	15	27	27	4
16:45:09	1	16	28	28	5
16:45:11	1	17	28	29	6
16:45:14	1	16	28	31	7
16:45:16	1	16	28	32	8
16:45:18	1	16	27	33	9
16:45:20	1	17	27	34	10
16:45:22	1	17	27	35	11
16:45:24	1	18	27	36	12

WAKTU	ID NODE	KECEPATAN ANGIN (MPH)	SUHU UDARA (CELCIUS)	NO URUT END DEVICE	NO URUT KOORDINATOR
16:45:28	1	2	27	37	13
16:45:30	1	2	27	38	14
16:45:32	1	4	27	39	15
16:45:34	1	16	27	40	16
16:45:36	1	18	27	41	17
16:45:38	1	19	28	42	18
16:45:40	1	2	28	43	19
16:45:43	1	2	29	44	20
16:45:45	1	2	29	45	21
16:45:47	1	2	28	46	22
16:45:49	1	16	27	47	23
16:45:51	1	16	27	48	24
16:45:53	1	16	27	49	25

5.2.1.2 Uji Coba pada Jarak 10 Meter

Gambar 5.8 adalah visualisasi jarak antara ZigBee *end device* dan ZigBee koordinator.



Gambar 5.8 Visualisasi jarak ZigBee *end device* dan koordinator

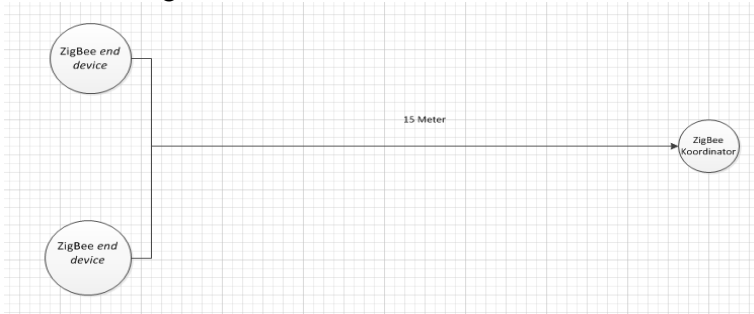
Pada jarak 10 meter, kesalahan sistem sedikit bertambah, pada data urut nomor 23, 26, 30, dan 32 data yang dikirim oleh *end device* hilang. Maka, jumlah jumlah *packet loss* pada sistem ialah 4. Tabel 5.5 adalah hasil uji coba pada jarak 10 meter.

Tabel 5.5 Data uji coba jarak 10 meter

WAKTU	ID NODE	KECEPATAN ANGIN (MPH)	SUHU UDARA (CELCIUS)	NO URUT END DEVICE	NO URUT KOORDI- NATOR
16:42:04	1	8	28	12	1
16:42:06	1	3	27	13	2
16:42:09	1	5	27	14	3
16:42:11	1	7	27	15	4
16:42:13	1	9	28	16	5
16:42:15	1	12	28	17	6
16:42:17	1	15	28	18	7
16:42:19	1	13	27	19	8
16:42:22	1	10	27	20	9
16:42:25	1	12	27	21	10
16:42:27	1	13	27	22	11
16:42:29	1	15	27	24	12
16:42:32	1	14	28	25	13
16:42:34	1	13	28	27	14
16:42:37	1	12	29	28	15
16:42:39	1	11	29	29	16
16:42:41	1	12	29	31	17
16:42:44	1	12	28	33	18
16:42:46	1	12	28	34	19
16:42:49	1	14	28	35	20
16:42:51	1	15	29	36	21
16:42:53	1	16	29	37	22
16:42:56	1	11	29	38	23
16:42:58	1	12	29	39	24
16:43:01	1	11	28	40	25

5.2.1.3 Uji Coba pada Jarak 15 Meter

Gambar 5.9 adalah visualisasi jarak antara ZigBee *end device* dan ZigBee koordinator.



Gambar 5.9 Visualisasi jarak ZigBee *end device* dan koordinator

Pada jarak 15 meter, kesalahan sistem jauh meningkat. Terlihat pada nomor urut paket *end device*, data nomor 14, 15, 18 dan 28 hilang. Maka, jumlah *packet loss* pada sistem ialah 4. Tabel 5.6 adalah data uji coba pada 15 meter.

Tabel 5.6 Data uji coba jarak 15 meter

WAKTU	ID NODE	KECEPATAN ANGIN (MPH)	SUHU UDARA (CELCIUS)	NO URUT END DEVICE	NO URUT KOORDI-NATOR
16:37:34	1	15	27	7	1
16:37:36	1	14	28	8	2
16:37:38	1	17	28	9	3
16:37:41	1	17	29	10	4
16:37:42	1	16	29	11	5
16:37:44	1	16	28	12	6
16:37:46	1	18	28	13	7
16:37:48	1	17	27	16	8
16:37:51	1	19	27	17	9
16:37:53	1	17	27	19	10
16:37:55	1	18	28	20	11

WAKTU	ID NODE	KECEPATAN ANGIN (MPH)	SUHU UDARA (CELCIUS)	NO URUT END DEVICE	NO URUT KOORDI- NATOR
16:37:57	1	15	27	21	12
16:37:59	1	17	27	22	13
16:38:01	1	16	27	23	14
16:38:04	1	19	27	24	15
16:38:06	1	19	27	25	16
16:38:08	1	17	27	26	17
16:38:11	1	18	27	27	18
16:38:13	1	18	28	29	19
16:38:15	1	17	28	30	20
16:38:17	1	16	29	31	21
16:38:20	1	16	29	32	22
16:38:22	1	15	28	33	23
16:38:25	1	14	27	34	24
16:38:28	1	17	27	35	25

5.2.2 Uji Coba Sistem pada Lapangan Sepak Bola ITS

Uji coba sistem pada lapangan sepak bola ITS ini saya menempatkan ZigBee koordinator dan ZigBee *end device* pada berbagai jarak, yaitu jarak 5 meter, 15 meter, 30 meter, 70 meter, dan 180 meter. Jarak dipilih karena dianggap cukup mewakili berbagai range jarak di sekitarnya. Skenario uji coba ini bisa dilihat pada Tabel 5.7.

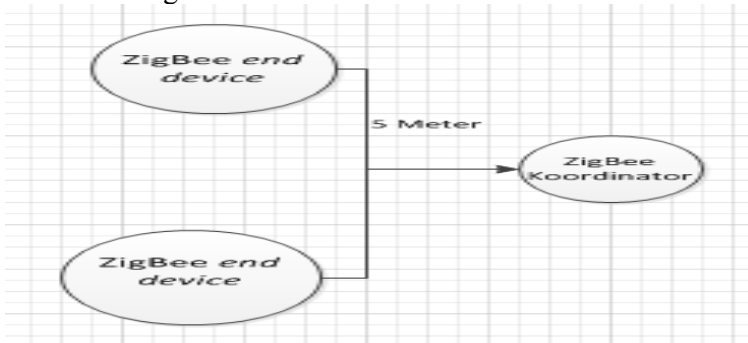
Tabel 5.7 Skenario uji coba pada lapangan sepak bola ITS

Nama	Uji coba pada lantai 4 Teknik Informatika ITS
Tujuan	Membandingkan data yang diterima pada berbagai jarak
Skenario	ZigBee <i>end device</i> dan ZigBee koordinator ditempatkan pada berbagai jarak

Hasil Uji Coba	Data berhasil didapat dan dibandingkan
-----------------------	--

5.2.2.1 Uji Coba Sistem pada Jarak 5 Meter

Gambar 5.10 adalah visualisasi jarak antara ZigBee *end device* dan ZigBee koordinator.



Gambar 5.10 Visualisasi jarak ZigBee *end device* dan koordinator

Pada jarak ini, data yang diterima oleh ZigBee koordinator tergolong baik. Terdapat *packet loss* pada jarak ini, data nomor 37 dan 44 pada nomor urut *end device* hilang. Tabel 5.8 adalah hasil uji coba pada 5 meter.

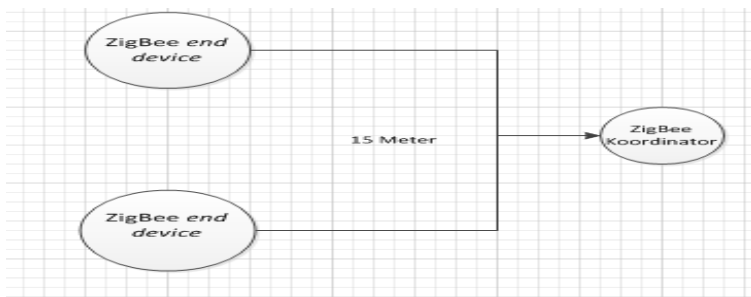
Tabel 5.8 Data uji coba jarak 5 meter

WAKTU	ID NODE	KECEPATAN ANGIN (MPH)	SUHU UDARA (CELCIUS)	NO URUT END DEVICE	NO URUT KOORDI-NATOR
15:21:17	1	2	30	19	1
15:21:19	1	5	31	20	2
15:21:22	1	7	31	21	3
15:21:25	1	6	32	22	4
15:21:27	1	4	32	23	5
15:21:29	1	6	32	24	6
15:21:31	1	8	32	25	7
15:21:34	1	7	31	26	8
15:21:36	1	7	31	27	9

WAKTU	ID NODE	KECEPATAN ANGIN (MPH)	SUHU UDARA (CELCIUS)	NO URUT END DEVICE	NO URUT KOORDI- NATOR
15:21:38	1	8	31	28	10
15:21:41	1	9	32	29	11
15:21:43	1	7	32	30	12
15:21:45	1	6	33	31	13
15:21:47	1	4	33	32	14
15:21:49	1	2	33	33	15
15:21:51	1	5	33	34	16
15:21:53	1	7	33	35	17
15:21:56	1	8	32	36	18
15:21:58	1	6	32	38	19
15:22:01	1	7	32	39	20
15:22:03	1	4	32	40	21
15:22:05	1	5	32	41	22
15:22:07	1	8	31	42	23
15:22:10	1	6	31	43	24
15:22:12	1	7	31	45	25

5.2.2.2 Uji Coba Sistem pada Jarak 15 Meter

Gambar 5.11 adalah visualisasi jarak antara ZigBee *end device* dan ZigBee koordinator.



Gambar 5.11 Visualisasi jarak ZigBee *end device* dan koordinator

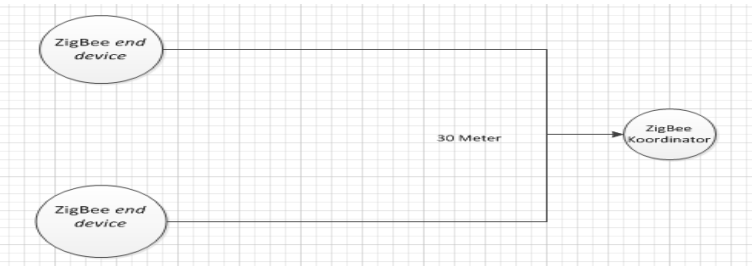
Pada jarak ini, hasil yang diterima oleh ZigBee koordinator mengalami penurunan. Data nomor 40, 44, dan 46 yang dikirim oleh ZigBee *end device* hilang. Tabel 5.9 adalah hasil uji coba pada jarak 15 meter.

Tabel 5.9 Data uji coba jarak 15 meter

WAKTU	ID NODE	KECEPATAN ANGIN (MPH)	SUHU UDARA (CELCIUS)	NO URUT END DEVICE	NO URUT KOORDI- NATOR
15:29:03	1	3	30	31	1
15:29:05	1	2	31	32	2
15:29:08	1	4	32	33	3
15:29:10	1	2	32	34	4
15:29:13	1	4	32	35	5
15:29:15	1	6	32	36	6
15:29:17	1	7	32	37	7
15:29:19	1	5	31	38	8
15:29:22	1	4	31	39	9
15:29:24	1	3	31	41	10
15:29:26	1	4	32	42	11
15:29:28	1	4	32	43	12
15:29:31	1	5	31	45	13
15:29:33	1	6	31	47	14
15:29:35	1	3	31	48	15
15:29:38	1	5	32	49	16
15:29:40	1	4	32	50	17
15:29:42	1	6	32	51	18
15:29:44	1	5	31	52	19
15:29:46	1	7	31	53	20
15:29:49	1	8	32	54	21
15:29:51	1	10	32	55	22
15:29:54	1	7	33	56	23
15:29:56	1	6	33	57	24
15:29:58	1	9	33	58	25

5.2.2.3 Uji Coba Sistem pada Jarak 30 Meter

Gambar 5.12 adalah visualisasi jarak antara ZigBee *end device* dan ZigBee koordinator.



Gambar 5.12 Visualisasi jarak ZigBee *end device* dan koordinator

Pada jarak 30 meter, *packet loss* meningkat. Data nomor 31, 41, 44 dan 45 yang dikirim oleh ZigBee *end device* hilang. Maka, jumlah kesalahan data pada sistem ialah 4. Tabel 5.10 adalah data uji coba dengan jarak 30 meter.

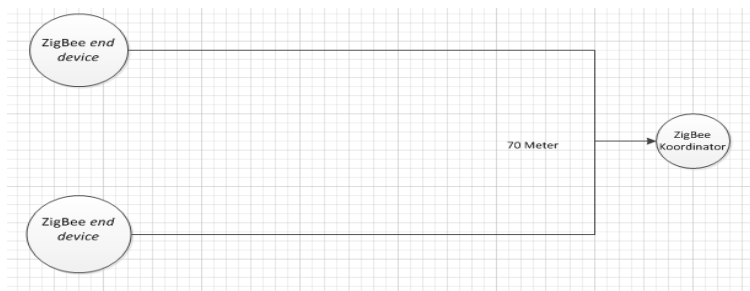
Tabel 5.10 Data uji coba jarak 30 meter

WAKTU	ID NODE	KECEPATAN ANGIN (MPH)	SUHU UDARA (CELCIUS)	NO URUT END DEVICE	NO URUT KOORDI- NATOR
15:37:17	1	7	30	23	1
15:37:17	1	9	31	24	2
15:37:17	1	11	32	25	3
15:37:17	1	10	32	26	4
15:37:17	1	8	32	27	5
15:37:17	1	6	32	28	6
15:37:17	1	7	32	29	7
15:37:17	1	9	32	30	8
15:37:17	1	8	32	32	9
15:37:17	1	8	31	33	10
15:37:17	1	10	31	34	11
15:37:17	1	9	31	35	12

WAKTU	ID NODE	KECEPATAN ANGIN (MPH)	SUHU UDARA (CELCIUS)	NO URUT END DEVICE	NO URUT KOORDI- NATOR
15:37:17	1	9	30	36	13
15:37:17	1	7	31	37	14
15:37:17	1	6	32	38	15
15:37:17	1	8	32	39	16
15:37:17	1	6	32	40	17
15:37:17	1	7	31	42	18
15:37:17	1	9	31	43	19
15:37:17	1	11	31	46	20
15:37:17	1	8	30	47	21
15:37:17	1	2	30	48	22
15:37:17	1	7	31	50	23
15:37:17	1	9	32	51	24
15:37:17	1	9	33	52	25

5.2.2.4 Uji Coba Sistem pada Jarak 70 Meter

Gambar 5.13 adalah visualisasi jarak antara ZigBee *end device* dan ZigBee koordinator.



Gambar 5.13 Visualisasi jarak ZigBee *end device* dan koordinator

Pada jarak 70 meter, tingkat kesalahan sistem bertambah. Data nomor 51, 56, 62, 63, 69, 71, 72, dan 73 yang dikirim oleh ZigBee *end device* hilang. Maka, jumlah kesalahan data pada

sistem ialah 8. Tabel 5.11 adalah data uji coba dengan jarak 70 meter.

Tabel 5.11 Data uji coba jarak 70 meter

WAKTU	ID NODE	KECEPATAN ANGIN (MPH)	SUHU UDARA (CELCIUS)	NO URUT END DEVICE	NO URUT KOORDI- NATOR
15:46:23	1	5	31	42	1
15:46:25	1	4	32	43	2
15:46:27	1	7	32	44	3
15:46:30	1	8	32	45	4
15:46:32	1	10	32	46	5
15:46:34	1	11	32	47	6
15:46:37	1	9	31	48	7
15:46:39	1	6	31	49	8
15:46:41	1	8	30	50	9
15:46:43	1	10	30	52	10
15:46:46	1	12	30	53	11
15:46:48	1	13	31	54	12
15:46:50	1	14	31	55	13
15:46:52	1	12	30	57	14
15:46:54	1	10	30	58	15
15:46:56	1	7	31	59	16
15:46:58	1	6	31	60	17
15:47:01	1	9	30	61	18
15:47:03	1	8	31	64	19
15:47:05	1	8	31	65	20
15:47:07	1	9	32	66	21
15:47:09	1	7	32	67	22
15:47:12	1	8	33	68	23
15:47:14	1	10	32	70	24
15:47:16	1	8	31	74	25

5.2.2.5 Uji Coba Sistem pada Jarak 180 Meter

Gambar 5.14 adalah visualisasi jarak antara ZigBee *end device* dan ZigBee koordinator.



Gambar 5.14 Visualisasi jarak ZigBee *end device* dan koordinator

Pada uji coba jarak 180 meter, data nomor 20, 26, 31, 34, 47, 38, 40, 41, 48, 49, 50, 51, 52, 53, dan 54 yang dikirim oleh ZigBee *end device* hilang. Maka, jumlah kesalahan data pada sistem ialah 15. Tabel 5.12 adalah hasil uji coba pada jarak 180 meter.

Tabel 5.12 Data uji coba jarak 180 meter

WAKTU	ID NODE	KECEPATAN ANGIN (MPH)	SUHU UDARA (CELCIUS)	NO URUT END DEVICE	NO URUT KOORDI- NATOR
16:03:07	1	9	31	17	1
16:03:09	1	11	30	18	2
16:03:11	1	13	31	19	3
16:03:13	1	11	31	21	4
16:03:15	1	14	32	22	5
16:03:18	1	15	31	23	6
16:03:20	1	13	31	24	7
16:03:22	1	12	32	25	8
16:03:25	1	10	32	27	9
16:03:27	1	12	31	28	10
16:03:29	1	11	31	29	11

WAKTU	ID NODE	KECEPATAN ANGIN (MPH)	SUHU UDARA (CELCIUS)	NO URUT END DEVICE	NO URUT KOORDI- NATOR
16:03:32	1	13	31	30	12
16:03:34	1	15	31	32	13
16:03:36	1	13	32	33	14
16:03:38	1	11	31	35	15
16:03:40	1	9	32	36	16
16:03:42	1	11	31	39	17
16:03:44	1	12	30	42	18
16:03:47	1	14	31	43	19
16:03:50	1	15	30	44	20
16:03:53	1	13	31	45	21
16:03:56	1	11	31	46	22
16:03:58	1	9	31	47	23
16:04:01	1	7	32	50	24
16:04:04	1	9	31	55	25

5.2.3 Perbandingan Berbagai Jarak dan Lokasi

Kali ini akan dibandingkan data dari berbagai jarak dan lokasi yang telah diuji cobakan sebelumnya.

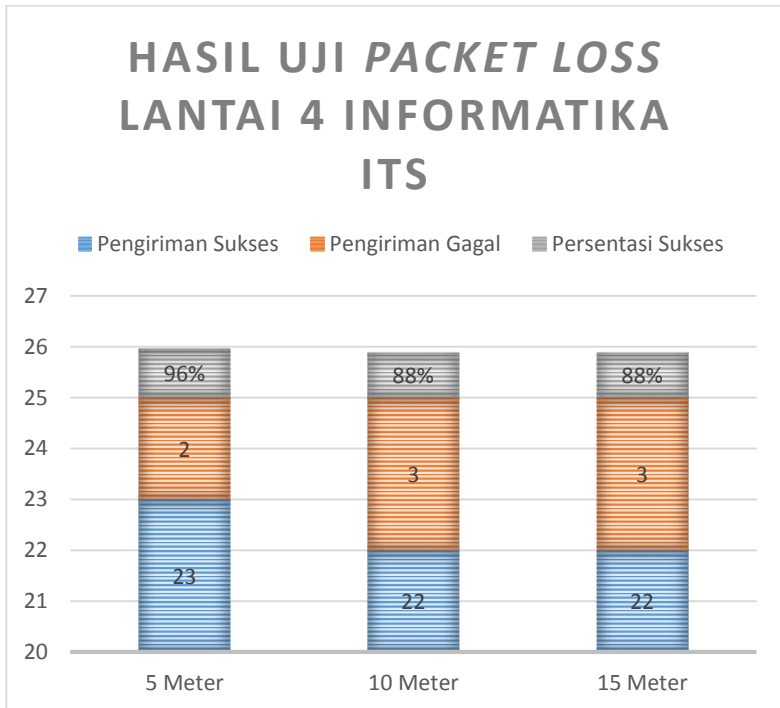
5.2.3.1 Perbandingan Uji Coba di Lantai 4 Teknik Informatika ITS

Tabel 5.13 adalah hasil dari perbandingan 25 data dengan berbagai jarak di lantai 4 Teknik Informatika ITS.

Tabel 5.13 Hasil uji *packet loss* lantai 4 Teknik Informatika ITS

Jarak	Pengiriman Sukses	Pengiriman Gagal
5 meter	24 kali	1 kali
10 meter	21 kali	4 kali
15 meter	21 kali	4 kali

Selanjutnya, Gambar 5.15 adalah grafik yang menunjukkan bahwa pada jarak 5 meter, *packet loss* berjumlah 1 paket, sedangkan pada jarak 10 dan 15 meter, *packet loss* berjumlah 4 paket.



Gambar 5.15 Grafik hasil pengujian *packet loss* lantai 4 Teknik Informatika ITS

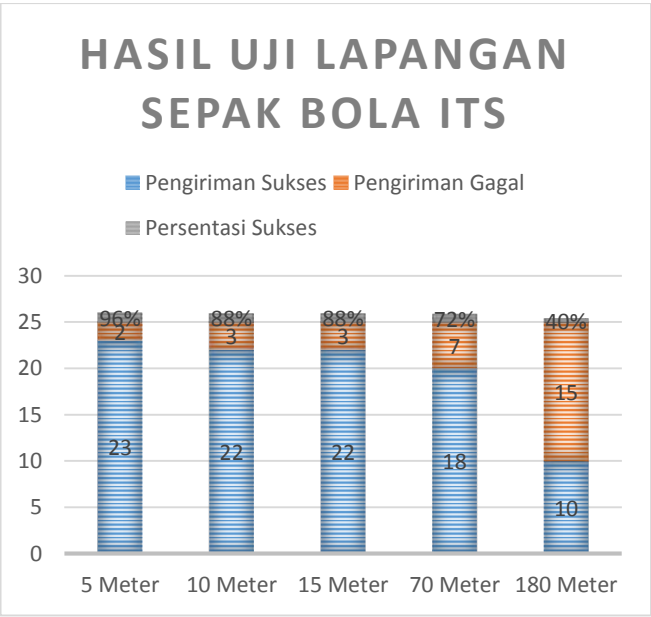
5.2.3.2 Perbandingan Uji Coba di Lapangan Sepak Bola ITS

Tabel 5.14 adalah hasil dari perbandingan 25 data dengan perbandingan jarak di lapangan sepak bola ITS.

Tabel 5.14 Hasil pengujian *packet loss* di lapangan sepak bola ITS

Jarak	Pengiriman Sukses	Pengiriman Gagal
5 meter	23 kali	2 kali
15 meter	22 kali	3 kali
30 meter	21 kali	4 kali
70 meter	20 kali	5 kali
180 meter	10 kali	15 kali

Selanjutnya, Gambar 5.16 akan memperlihatkan grafik hasil dari pengujian *packet loss* pada lapangan sepak bola ITS. Dapat dilihat pada grafik, *packet loss* yang paling rendah terdapat pada jarak 5 meter, pengiriman gagal berjumlah 2 buah, dan yang paling tinggi ialah pada jarak 180 meter, yaitu berjumlah 15 buah.



Gambar 5.16 Grafik hasil pengujian *packet loss* lapangan sepak bola ITS

Dari hasil uji coba pada lantai 4 Teknik Informatika ITS dan lapangan sepak bola ITS dapat diambil kesimpulan bahwa jumlah *packet loss* semakin meningkat dengan menaiknya jarak antara ZigBee koordinator dengan ZigBee *end device*. Hal ini dapat menjadi perhitungan untuk meletakkan ZigBee pada posisi tertentu.

Packet loss pada lantai 4 Teknik Informatika ITS bisa saja terjadi karena adanya halangan seperti tembok dan meja. Sedangkan pada lapangan sepak bola ITS sangat minim akan halangan terhadap sinyal ZigBee

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini berisi kesimpulan dan saran yang dapat diambil dari Tugas Akhir ada sebagai berikut.

6.1 Kesimpulan

Berdasarkan hasil pengamatan yang telah dilakukan dari tahap perancangan, kemudian implementasi, hingga uji coba aplikasi dapat diambil kesimpulan yaitu:

1. Semakin jauh jarak antara ZigBee koordinator dengan ZigBee *end device* dapat menyebabkan peningkatan kesalahan pengiriman data pada sistem.
2. Jarak ideal antara ZigBee koordinator dan *end device* ialah kurang lebih 70 meter di ruang terbuka. Jarak ideal merupakan jarak yang mempunyai tingkat kesalahan kecil, namun memiliki jangkauan yang cukup jauh.
3. Pemantauan cuaca dapat dilakukan secara *remote* melalui aplikasi *web* yang telah dibuat. Aplikasi *web* sistem monitoring akan menampilkan data kecepatan angin dan suhu udara yang telah ditangkap perangkat ZigBee.

6.2 Saran

Selama proses pengerjaan Tugas Akhir yang meliputi perancangan, implementasi dan uji coba, ditemukan beberapa kekurangan pada sistem monitoring suhu. Ada beberapa saran yang dapat dilakukan untuk mengembangkan lebih lanjut sistem ini yaitu:

1. Pengembangan ZigBee monitoring kecepatan angin dan suhu udara nantinya dapat dikembangkan menjadi perangkat untuk memantau keadaan cuaca pada daerah terpencil apabila diletakkan dengan jumlah perangkat ZigBee yang lebih banyak.

2. Semakin jauh jarak antara ZigBee koordinator dan ZigBee *end device* dapat menyebabkan peningkatan kesalahan pengiriman data pada sistem.
3. ZigBee tidak cocok untuk pengiriman jarak jauh yang memerlukan ketepatan data tinggi. Ke depannya akan lebih baik jika sistem diperbarui media pengirimannya dengan sesuatu yang lebih reliable seperti GSM *Shield*.

LAMPIRAN

1. Kode sumber fungsi *loop* pada ZigBee koordinator.

```

#define SOP '<'
#define EOP '>'

bool started = false;
bool ended = false;

char inData[64];
byte index;

void setup()
{
    Serial.begin(9600);
}

void loop()
{
    while(Serial.available() > 0)
    {
        char inChar = Serial.read();
        if(inChar == SOP)
        {
            index = 0;
            inData[index] = '\0';
            started = true;
            ended = false;
        }
        else if(inChar == EOP)
        {
            ended = true;
            break;
        }
        else
        {
            if(index < 63)
            {
                inData[index] = inChar;
                index++;
                inData[index] = '\0';
            }
        }
    }
    if(started && ended)
    {

```

```

        int values[2]; // Array to hold the values
        byte index = 0; // index into array

        char *token = strtok(inData, ",");
        while(token)
        {
            values[index] = atoi(token);
            index++; // Increment the index

            token = strtok(NULL, ","); string
        }
        Serial.print(values[0]);
        Serial.print(" ");
        Serial.print(values[1]);
        Serial.print(" ");
        Serial.print(values[2]);
        Serial.print(" ");
        Serial.print("\n");
        started = false;
        ended = false;
        index = 0;
        inData[index] = '\0';
    }

    delay(1000);
}

```

2. Kode sumber fungsi untuk mengirimkan data ke ZigBee *end device* dari ZigBee koordinator.

```
#include <DHT.h>
#define DHTPIN 7
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);

char dataPacket[64];
long pulsewidth1;
int wspeed1;
int temp;
void setup(){

  Serial.begin(9600);
  pinMode(8,INPUT);
  dht.begin();
}

void loop(){
  temp = dht.readTemperature();
  //int sensorValue = analogRead(A0);
  int id = 1;
  //sensorValue = (sensorValue * 3)/10;
  pulsewidth1=pulseIn(8,LOW);
  wspeed1=1000000.0/pulsewidth1*2.5;
  if(wspeed1 > 500){
    wspeed1 = 2;
  }
  else if(wspeed1 < 1){
    wspeed1 = 2.4;
  }
  sprintf(dataPacket,"%d,%d,%d",id, wspeed1, temp);
  Serial.println(dataPacket);
  delay(1000);
}
```

3. Kode sumber fungsi untuk memasukkan data ke database

```

import serial
import time
import MySQLdb as mdb
import sys

#establish connection to MySQL. You'll have to change this for
    your database.
dbConn = mdb.connect("localhost","root","", "tugasakhir") or
    die ("could not connect to database")
#open a cursor to the database
#cursor = dbConn.cursor()

device = 'COM14' #this will have to be changed to the serial port
    you are using
try:
print "Trying...",device
arduino = serial.Serial(device, 9600)
except:
print "Failed to connect on",device
try:
while 2:
cursor = dbConn.cursor()
time.sleep(2)
data = arduino.readline() #read the data from the arduino
pieces = data.split(" ") #split the data by the tab
print pieces
#Here we are going to insert the data into the Database
try:

```



```
#print 56 #verify the execution in the Try-loop
cursor.execute      ("INSERT      INTO      datanode
                    (id_pengirim,kecepatan_angin,suhu_udara)
                    VALUES      (%s,%s,%s)",
                    (pieces[0],pieces[1],pieces[2]))
print 5    #verify the execution in the Try-loop
dbConn.commit() #commit the insert
cursor.close() #close the cursor
except MySQLdb.IntegrityError:
print "failed to insert data"
finally:
cursor.close() #close just incase it failed
except:
print "Failed to get data from Arduino!"
```

DAFTAR PUSTAKA

- [1] Arduino, "Arduino-Introduction," Arduino, 12 10 2013. [Online]. Available: <http://arduino.cc/en/Guide/Introduction>. [Accessed 14 11 2013].
- [2] M. Specialities, "Datasheet LTD Series," 2011.
- [3] "Dfrobot," [Online]. Available: www.dfrobot.com. [Accessed 29 Oktober 2014].
- [4] "PHP," [Online]. Available: <http://php.net/>. [Diakses 30 April 2013].
- [5] B. Banjarbaru, "Angin Puting Beliung," 15 12 2012. [Online]. Available: <http://klimatologibanjarbaru.com/lain-lain/artikel/puting-beliung/>. [Accessed 10 01 2015].
- [6] "Jenis-Jenis Angin," [Online]. Available: <http://idkf.bogor.net/yuesbi/e-DU.KU/edukasi.net/Fenomena.Alam/Jenis.Angin/semua.html>. [Accessed 10 01 2015].
- [7] "W3 School," [Online]. Available: http://www.w3schools.com/php/php_intro.asp. [Diakses 30 April 2013].